



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.413

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(04/95)

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

**TECHNOLOGIES DE L'INFORMATION –
SYSTÈMES DE MESSAGERIE:
MÉMOIRE DE MESSAGES:
DÉFINITION DU SERVICE ABSTRAIT**

Recommandation UIT-T X.413

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

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

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

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.413 de l'UIT-T a été approuvé le 10 avril 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10021-5.

NOTE

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

© UIT 1997

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.

RECOMMANDATIONS UIT-T DE LA SÉRIE X

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

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

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

TABLE DES MATIÈRES

	<i>Page</i>
SECTION 1 – GÉNÉRALITÉS	1
1 Champ d'application.....	1
2 Références normatives	1
2.1 Références relatives au modèle de référence	2
2.2 Références relatives à la présentation	2
2.3 Références relatives aux opérations distantes	2
2.4 Références relatives à l'annuaire	2
2.5 Références relatives à la messagerie	2
3 Définitions.....	3
3.1 Définitions communes pour le système de messagerie.....	3
3.2 Définitions concernant l'enregistrement des messages	3
4 Abréviations	7
5 Conventions.....	8
5.1 Conventions applicables aux services abstraits.....	8
5.2 Conventions utilisées dans les Tableaux 2 et 3 de l'article 11 et applicables aux types d'attributs.....	8
5.3 Conventions applicables aux types d'attributs utilisés dans le Tableau 4 de l'article 11	9
5.4 Conventions générales d'écriture	9
5.5 Conventions d'écriture applicables aux définitions ASN.1	9
5.6 Règles applicables aux définitions ASN.1	9
5.7 Conventions applicables aux éditions antérieures de la présente Définition de service	9
SECTION 2 – DÉFINITION DU SERVICE ABSTRAIT D'ENREGISTREMENT DE MESSAGES	10
6 Modèle de mémoire de messages (MS).....	10
6.1 Contrats et objets de la mémoire de messages	11
6.2 Accès de la mémoire de messages	12
6.2.1 Accès d'extraction	12
6.2.2 Accès de dépôt MS	12
6.2.3 Accès d'administration	13
6.3 Modèle informationnel.....	13
6.3.1 Classes d'entrée	13
6.3.2 Entrées	14
6.3.3 Attributs	14
6.3.4 Entrées principales, entrées mères et entrées filles	16
6.3.5 Attributs propres au contenu	17
6.3.6 Types d'entrée	17
6.3.7 Organisation des classes d'entrée	18
6.3.8 Etat d'extraction	20
6.3.9 Règles de criblage	21
6.4 Groupage des messages	23
6.5 Actions automatiques.....	24
6.5.1 Classe d'objets informationnels AUTO-ACTION (actions automatiques)	24
6.5.2 Consignation d'action automatique	25
6.5.3 Erreurs d'action automatique.....	25
6.5.4 Exécution d'actions automatiques	25
6.6 Extensions MS	26
7 Opérations de rattachement et de détachement MS.....	26

	<i>Page</i>	
7.1	Opération abstraite MS-bind (rattachement MS).....	26
7.1.1	Argument de rattachement MS	27
7.1.2	Résultat du rattachement MS	29
7.1.3	Erreur de rattachement MS	31
7.2	Opération abstraite MS-unbind (détachement MS).....	31
8	Opérations abstraites	32
8.1	Types de données communs utilisés dans les opérations abstraites	32
8.1.1	Intervalle	32
8.1.2	Filtres	33
8.1.3	Sélecteur.....	35
8.1.4	Sélection d'information d'entrée.....	36
8.1.5	Information d'entrée	36
8.1.6	Options de dépôt MS	37
8.1.7	Résultats communs de dépôt.....	38
8.2	Opérations abstraites de l'accès d'extraction	38
8.2.1	Opération abstraite Summarize (récapitulation)	38
8.2.2	Opération abstraite List (listage).....	40
8.2.3	Opération abstraite Fetch (recherche)	41
8.2.4	Opération abstraite Delete (suppression)	43
8.2.5	Opération abstraite Register-MS (consignation MS)	44
8.2.6	Opération abstraite Alert (avertissement)	50
8.2.7	Opération abstraite Modify (modification)	51
8.3	Opérations abstraites de l'accès de dépôt MS	53
8.3.1	Opération abstraite MS-message-submission (dépôt de message MS).....	53
8.3.2	Opération abstraite MS-probe-submission (dépôt de message d'essai MS).....	55
8.3.3	Opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS)....	56
8.3.4	Opération abstraite MS-submission-control (commande de dépôt MS).....	56
9	Erreurs abstraites	57
9.1	Priorité des erreurs	57
9.2	Erreur d'attribut	57
9.3	Erreur de demande d'action automatique	58
9.4	Erreur de suppression.....	59
9.5	Erreur de restriction de recherche	59
9.6	Erreur de paramètres non valides.....	60
9.7	Erreur d'intervalle.....	60
9.8	Erreur de sécurité	60
9.9	Erreur de numéro de séquence	60
9.10	Erreur de service	61
9.11	Erreur de groupe de messages.....	61
9.12	Erreur d'extension MS.....	62
9.13	Erreur de consignation MS	63
9.14	Anciens pouvoirs mal spécifiés.....	63
9.15	Nouveaux pouvoirs inacceptables.....	63
9.16	Erreur de modification	63
9.17	Erreur de classe d'entrée.....	64
SECTION 3 – TYPES D'ATTRIBUTS GÉNÉRAUX, RÈGLES DE CRIBLAGE GÉNÉRALES ET TYPES D'ACTIONS AUTOMATIQUES GÉNÉRALES		65
10	Présentation générale.....	65
11	Types d'attributs généraux.....	65
11.1	Présentation générale des types d'attributs généraux	65
11.1.1	Prescriptions de prise en charge de types d'attributs généraux par la mémoire de messages	68
11.1.2	Prescriptions de prise en charge de types d'attributs généraux par l'utilisateur MS.....	68

	<i>Page</i>
11.2 Description des types d'attributs généraux	68
11.2.1 Liste des rapports corrélés AC (AC-correlated-report-list).....	69
11.2.2 Récapitulation des rapports AC (AC-report-summary)	69
11.2.3 Liste des rapports non corrélés AC (AC-uncorrelated-report-list).....	70
11.2.4 Erreur d'action automatique (auto-action-error).....	70
11.2.5 Identificateur de consignation d'action automatique (auto-action-registration-identifiant)	71
11.2.6 Entrée de l'objet de l'action automatique (auto-action-subject-entry)	71
11.2.7 Type d'action automatique (auto-action-type)	71
11.2.8 Numéros de séquence des entrées filles (child-sequence-numbers)	71
11.2.9 Contenu (content).....	71
11.2.10 Identificateur d'algorithme de confidentialité du contenu (content-confidentiality- algorithm-identifiant).....	72
11.2.11 Corrélateur de contenu (content-correlator).....	72
11.2.12 Identificateur de contenu (content-identifiant)	72
11.2.13 Vérification d'intégrité du contenu (content-integrity-check).....	72
11.2.14 Longueur du contenu (content-length).....	72
11.2.15 Contenu renvoyé (content-returned)	72
11.2.16 Type de contenu (content-type)	73
11.2.17 Conversion avec perte interdite (conversion-with-loss-prohibited).....	73
11.2.18 Types d'information codée convertis (converted EITs)	73
11.2.19 Heure de création (creation-time)	73
11.2.20 Heure d'annulation de la remise différée (deferred-delivery-cancellation-time)	73
11.2.21 Heure de remise différée (deferred-delivery-time)	74
11.2.22 Heure de suppression (deletion-time)	74
11.2.23 Types d'information codée remis (delivered EITs).....	74
11.2.24 Fanions de remise (delivery-flags).....	74
11.2.25 Chronologie du développement de la liste de distribution (DL-expansion-history)	75
11.2.26 Interdiction du développement de la liste de distribution (DL-expansion-prohibited)	75
11.2.27 Type d'entrée (entry-type).....	75
11.2.28 Information de trace interne (internal-trace-information).....	75
11.2.29 Heure limite de remise (latest-delivery-time)	75
11.2.30 Marqué pour être supprimé (marked-for-deletion)	76
11.2.31 Enveloppe de remise de message (message-delivery-envelope).....	76
11.2.32 Heure de remise du message (message-delivery-time).....	76
11.2.33 Nom de groupe de messages (message-group-name)	76
11.2.34 Identificateur de message (message-identifiant).....	77
11.2.35 Notes de message (message-notes)	77
11.2.36 Vérification d'authentification de l'origine du message (message-origin-authentication- check).....	77
11.2.37 Etiquette de sécurité du message (message-security-label)	77
11.2.38 Enveloppe de dépôt de message (message-submission-envelope).....	77
11.2.39 Heure de dépôt du message (message-submission-time).....	78
11.2.40 Jeton de message (message-token).....	78
11.2.41 Origine MS (MS-originated).....	78
11.2.42 Erreur de dépôt MS (MS-submission-error)	78
11.2.43 Types d'information codée d'origine (original-EITs).....	79
11.2.44 Nom de destinataire prévu au départ (originally-intended-recipient-name)	79
11.2.45 Certificat de l'agent MTA expéditeur (originating-MTA-certificate)	79
11.2.46 Certificat de l'expéditeur (originator-certificate).....	79
11.2.47 Nom de l'expéditeur (originator-name).....	79
11.2.48 Demande de rapport par l'expéditeur (originator-report-request)	80
11.2.49 Adresse de retour à l'expéditeur (originator-return-address)	80
11.2.50 Noms d'autres destinataires (other-recipient-names)	80
11.2.51 Numéro de séquence d'entrée mère (parent-sequence-number).....	80
11.2.52 Indicateurs par message (per-message-indicators).....	80
11.2.53 Champs de dépôt de message par destinataire (per-recipient-message- submission-fields).....	81

11.2.54	Champs de dépôt de message d'essai par destinataire (per-recipient-probe-submission-fields)	81
11.2.55	Champs de dépôt de rapport par destinataire (per-recipient-report-delivery-fields)	81
11.2.56	Priorité (priority)	81
11.2.57	Vérification d'authentification de l'origine du message d'essai (probe-origin-authentication-check)	81
11.2.58	Enveloppe de dépôt de message d'essai (probe-submission-envelope)	81
11.2.59	Demande de preuve de remise (proof-of-delivery-request)	82
11.2.60	Preuve de dépôt (proof-of-submission).....	82
11.2.61	Noms des destinataires (recipient-names).....	82
11.2.62	Interdiction de réaffectation de destinataire (recipient-reassignment-prohibited)	82
11.2.63	Chronologie du réacheminement (redirection-history)	82
11.2.64	Enveloppe de remise de rapport (report-delivery-envelope).....	83
11.2.65	Nom de la liste de distribution rapporteur (reporting-DL-name).....	83
11.2.66	Certificat de l'agent MTA rapporteur (reporting-MTA-certificate)	83
11.2.67	Vérification d'authentification de l'origine du rapport (report-origin-authentication-check).....	83
11.2.68	Etat d'extraction (retrieval-status)	83
11.2.69	Classification de sécurité (security-classification)	84
11.2.70	Numéro de séquence (sequence-number)	84
11.2.71	Période d'enregistrement (storage-period)	84
11.2.72	Heure d'enregistrement (storage-time).....	84
11.2.73	Identificateur de dépôt du sujet (subject-submission-identifiant).....	85
11.2.74	Nom de ce destinataire (this-recipient-name)	85
11.2.75	Information de trace (trace-information)	85
11.3	Ensemble d'objets informationnels (Attribute-table)	85
11.4	Création des attributs généraux	86
11.5	Abonnement aux types d'attributs	86
11.6	Types d'attributs généraux pouvant être modifiés.....	90
12	Règles de criblage générales	90
12.1	Syntaxe MS-string (chaîne MS).....	90
12.2	Règles de criblage basées sur les chaînes	91
12.2.1	Crible de chaînes MS	91
12.2.2	Crible d'ordonnancement de chaînes MS	91
12.2.3	Crible de sous-chaînes MS.....	92
12.2.4	Crible de sous-chaînes individuelles MS	92
12.2.5	Crible de chaînes MS sensible à la capitalisation	92
12.2.6	Crible de listes de chaînes MS	92
12.2.7	Crible d'éléments de listes de chaînes MS	93
12.2.8	Crible de listes de sous-chaînes individuelles MS	93
12.2.9	Crible d'éléments de listes de sous-chaînes individuelles MS	93
12.3	Règles de criblage syntaxiques	93
12.4	Règles de criblage pour les attributs complexes de la mémoire de messages.....	94
12.4.1	Crible d'adresse OR.....	94
12.4.2	Crible d'éléments d'adresse OR.....	95
12.4.3	Crible d'éléments de sous-chaînes d'adresse OR.....	95
12.4.4	Crible de nom OR	95
12.4.5	Crible d'éléments de nom OR	96
12.4.6	Crible d'éléments de sous-chaînes de nom OR	96
12.4.7	Crible d'éléments individuels de nom OR.....	96
12.4.8	Crible de réacheminement ou de développement de liste de distribution.....	97
12.4.9	Crible d'éléments du réacheminement ou du développement de liste de distribution.....	97
12.4.10	Crible d'éléments de sous-chaînes du réacheminement ou du développement de liste de distribution	97
12.4.11	Crible de motifs de réacheminement.....	97
12.4.12	Crible d'identificateur MTS	97
12.4.13	Crible de corrélateur de contenu	98
12.4.14	Crible d'identificateur de contenu	98

	<i>Page</i>	
12.5	Prise en charge des règles de criblage.....	98
12.6	Ensemble d'objets informationnels (Matching-rule-table).....	98
13	Actions automatiques générales.....	99
13.1	Avertissement automatique Auto-alert.....	100
13.2	Modification automatique (Auto-modify).....	101
13.3	Corrélation automatique des rapports (Auto-correlate-reports).....	102
13.4	Suppression automatique (Auto-delete).....	102
SECTION 4 – PROCÉDURES APPLICABLES À L'ENREGISTREMENT DE MESSAGES ET À LA RÉALISATION DES ACCÈS.....		103
14	Présentation générale.....	103
15	Utilisation du service abstrait du système de transfert de messages.....	103
15.1	Utilisation des services abstraits de l'accès de remise.....	104
15.1.1	Fonctionnement de l'opération abstraite Message-delivery (remise de message).....	104
15.1.2	Fonctionnement de l'opération abstraite Report-delivery (remise de rapport).....	104
15.1.3	Invocation de l'opération abstraite Delivery-control (commande de remise).....	105
15.1.4	Règles de création applicables aux attributs généraux.....	105
15.2	Utilisation des services abstraits de l'accès de dépôt.....	106
15.2.1	Invocation de l'opération abstraite Message-submission (dépôt de message).....	106
15.2.2	Invocation de l'opération abstraite Probe-submission (dépôt de message d'essai).....	106
15.2.3	Invocation de l'opération abstraite Cancel-deferred-delivery (annulation de remise différée).....	106
15.2.4	Fonctionnement de l'opération abstraite Submission-control (commande de dépôt).....	107
15.3	Utilisation des services abstraits de l'accès d'administration.....	107
15.3.1	Invocation de l'opération abstraite Register (consignation).....	107
15.3.2	Invocation de l'opération abstraite Change-credentials (modification des pouvoirs).....	107
15.3.3	Fonctionnement de l'opération abstraite Change-credentials (modification des pouvoirs).....	108
16	Fourniture du service abstrait de mémoire de messages.....	108
16.1	Fourniture des services abstraits de l'accès d'extraction.....	108
16.1.1	Fonctionnement de l'opération abstraite Summarize (récapitulation).....	108
16.1.2	Fonctionnement de l'opération abstraite List (listage).....	108
16.1.3	Fonctionnement de l'opération abstraite Fetch (recherche).....	109
16.1.4	Fonctionnement de l'opération abstraite Delete (suppression).....	110
16.1.5	Fonctionnement de l'opération abstraite Register-MS (consignation MS).....	110
16.1.6	Fonctionnement de l'opération abstraite Modify (modification).....	110
16.1.7	Invocation de l'opération abstraite Alert (avertissement).....	111
16.2	Fourniture des services abstraits de l'accès de dépôt MS.....	111
16.2.1	Fonctionnement de l'opération abstraite MS-message-submission (dépôt de message MS).....	111
16.2.2	Fonctionnement de l'opération abstraite MS-probe-submission (dépôt de message d'essai MS).....	113
16.2.3	Fonctionnement de l'opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS).....	114
16.2.4	Invocation de l'opération abstraite Submission-control (commande de dépôt).....	114
16.2.5	Règles de création des attributs généraux.....	114
16.3	Fourniture des services abstraits de l'accès d'administration.....	115
16.3.1	Fonctionnement de l'opération abstraite Register (consignation).....	115
16.3.2	Invocation de l'opération abstraite Change-credentials (modification des pouvoirs).....	115
16.3.3	Fonctionnement de l'opération abstraite Change-credentials (modification des pouvoirs).....	116
17	Réalisation des accès.....	116
17.1	Accès d'extraction.....	116
17.2	Accès de dépôt MS.....	116
17.3	Accès d'administration.....	116

	<i>Page</i>
Annexe A – Affectation formelle des identificateurs d'objets.....	117
Annexe B – Définition formelle du service abstrait de mémoire de messages.....	120
Annexe C – Définition formelle des types d'attributs généraux	134
Annexe D – Définition formelle des règles de criblage générales	144
Annexe E – Définition formelle des types d'actions automatiques générales	147
Annexe F – Récapitulation des modifications apportées aux précédentes éditions.....	149
Annexe G – Définition formelle des bornes supérieures des paramètres MS	151
Annexe H – Groupage des messages.....	153
Annexe I – Exemple d'opération abstraite Summarize (récapitulation)	155
Annexe J – Différences entre la Rec. UIT-T X.413 (1995) et l'ISO/CEI 10021-5:1995	156
INDEX.....	157

Résumé

La présente Recommandation comporte les quatorze nouveaux éléments de service concernant la mémoire de messages (MS) générale qui seront proposés dans la prochaine édition de la Recommandation F.400/X.400. Ces éléments élargissent la fonctionnalité de l'édition de 1992 de la Recommandation X.413 dans les domaines suivants:

- a) la mémoire de messages prend en charge l'enregistrement de messages déposés (auparavant, seuls les messages remis pouvaient être enregistrés). Les brouillons de messages peuvent également être enregistrés;
- b) un mécanisme de journalisation a été défini qui permet de consigner tous les événements de dépôt et de remise, et l'exécution d'actions automatiques choisies;
- c) le groupage des messages permet à l'utilisateur de la mémoire de messages de définir une hiérarchie d'enregistrement (similaire à la structure de répertoires de fichiers) afin de classer les messages enregistrés. Les messages peuvent être affectés manuellement ou automatiquement à des groupes de messages;
- d) les définitions ASN.1 ont été entièrement révisées afin de tirer parti des nouveaux outils offerts par les Recommandations X.680 et X.880 (Opérations distantes);
- e) des fonctionnalités ont été ajoutées afin d'améliorer la gestion des messages; on peut attacher manuellement ou automatiquement un attribut «durée de vie du message» à n'importe quel message. L'utilisateur de la mémoire de messages peut alors ordonner à ce service de supprimer automatiquement tout message dont la durée de vie est venue à expiration. Les messages peuvent être annotés manuellement ou automatiquement;
- f) les rapports peuvent être corrélés automatiquement avec les messages auxquels ils font référence, de sorte que l'utilisateur de la mémoire de messages puisse extraire facilement une récapitulation de tous les rapports de remise et de non-remise concernant un message particulier.

La Recommandation X.420 (1994) comporte sept autres nouveaux éléments de service spécifiques à la mémoire de messages pour la messagerie de personne à personne.

Introduction

La présente Recommandation | Norme internationale appartient à une série de Recommandations | Normes internationales qui définissent la messagerie dans un environnement de systèmes ouverts répartis.

La messagerie permet l'échange de messages entre utilisateurs par enregistrement et retransmission. Un message déposé par un utilisateur donné (l'expéditeur) est acheminé dans le système de transfert de messages (MTS) et remis à un ou à plusieurs autres utilisateurs (les destinataires).

La présente Recommandation | Norme internationale définit le service abstrait de mémoire de messages (service abstrait MS) qui permet l'extraction de messages à partir d'une mémoire de messages et le dépôt de messages dans un système de messagerie via la mémoire de messages. Le service abstrait de mémoire de messages fournit aussi des services de gestion de messages, définis par le service abstrait de système de transfert de messages (MTS).

La présente Recommandation | Norme internationale a été élaborée conjointement par l'UIT-T et l'ISO/CEI. La Norme internationale correspondante est l'ISO/CEI 10021-5:1995. L'Annexe J énumère les différences entre les deux documents.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – SYSTÈMES DE MESSAGERIE:
MÉMOIRE DE MESSAGES: DÉFINITION DU SERVICE ABSTRAIT**

SECTION 1 – GÉNÉRALITÉS

1 Champ d'application

La présente Recommandation | Norme internationale définit le service abstrait de mémoire de messages qui est assuré par le protocole d'accès à la mémoire de messages (spécifié dans la Rec. UIT-T X.419 | ISO/CEI 10021-6) conjointement avec le service abstrait de système de transfert de messages (défini dans la Rec. UIT-T X.411 | ISO/CEI 10021-4) ainsi qu'avec les services fournis par les éléments de service d'opérations distantes (ROSE) (définis dans la Rec. UIT-T X.219 | ISO/CEI 9072-1). La syntaxe abstraite des protocoles de la couche application utilisés dans la présente Recommandation | Norme internationale est définie dans la Rec. UIT-T X.680 | ISO/CEI 8824-1.

Les autres Recommandations | parties de Norme ISO/CEI 10021 définissent d'autres aspects des systèmes de messagerie. La Rec. UIT-T F.400/X.400 | ISO/CEI 10021-1 définit les services fournis aux utilisateurs par les systèmes de messagerie. La Rec. UIT-T X.402 | ISO/CEI 10021-2 donne un aperçu général de l'architecture des systèmes de messagerie. La Rec. UIT-T X.420 | ISO/CEI 10021-7 définit le service abstrait pour la messagerie de personne à personne ainsi que le format des messages de personne à personne.

La section 2 de la présente Recommandation | Norme internationale donne la définition du service abstrait de mémoire de messages. L'article 6 décrit le modèle du service abstrait de mémoire de messages. L'article 7 définit la sémantique et la syntaxe abstraite des opérations abstraites de rattachement et de détachement au service de mémoire de messages. L'article 8 définit la sémantique et la syntaxe abstraite des opérations du service abstrait de mémoire de messages (MS). L'article 9 définit la sémantique et la syntaxe abstraite des erreurs du service abstrait.

La section 3 de la présente Recommandation | Norme internationale définit les types d'attributs généraux, les règles de criblage générales et les types d'actions automatiques générales qui sont liés à l'enregistrement des messages. L'article 10 donne une présentation générale. L'article 11 définit la sémantique et la syntaxe abstraite des types d'attributs généraux. L'article 12 définit la sémantique et la syntaxe abstraite des règles de criblage générales. L'article 13 définit la sémantique et la syntaxe abstraite des types d'actions automatiques générales.

La section 4 de la présente Recommandation | Norme internationale décrit les procédures d'enregistrement des messages et la réalisation des accès. L'article 14 donne une présentation générale. L'article 15 décrit comment est assuré le service abstrait de système de transfert de messages (MTS). L'article 16 décrit comment est fourni le service abstrait de mémoire de messages. L'article 17 décrit comment sont réalisés les accès de la mémoire de messages.

Les critères de conformité à la présente Recommandation | Norme internationale sont précisés à l'article 10 de la Rec. UIT-T X.419 | ISO/CEI 10021-6.

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 Références relatives au modèle de référence

La présente Recommandation | Norme internationale cite la spécification suivante concernant le modèle de référence:

- 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.*

2.2 Références relatives à la présentation

La présente Recommandation | Norme internationale cite les spécifications suivantes concernant la présentation:

- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*

2.3 Références relatives aux opérations distantes

La présente Recommandation | Norme internationale cite la spécification suivante concernant les opérations distantes:

- Recommandation UIT-T X.880 (1994) | ISO/CEI 13712-1:1995, *Technologies de l'information – Opérations distantes: concepts, modèle et notation.*

2.4 Références relatives à l'annuaire

La présente Recommandation | Norme internationale cite les spécifications suivantes concernant l'annuaire:

- Recommandation UIT-T X.501 (1993) | ISO/CEI 9594-2:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles.*
- Recommandation UIT-T X.509 (1993) | ISO/CEI 9594-8:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre d'authentification.*
- Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés.*

2.5 Références relatives à la messagerie

La présente Recommandation | Norme internationale cite les spécifications suivantes concernant les systèmes de messagerie:

- Recommandation UIT-T F.400/X.400 (1993), *Services de messagerie – Système de messagerie et vue d'ensemble du service.*
ISO/CEI 10021-1:1994, *Technologies de l'information – Systèmes de messagerie (MHS) – Partie I: présentation générale du système et des services.*
- Recommandation UIT-T X.402 (1995) | ISO/CEI 10021-2:1996, *Technologies de l'information – Systèmes de messagerie: architecture globale.*
- Recommandation UIT-T X.411 (1995) | ISO/CEI 10021-4:1996, *Technologies de l'information – Systèmes de messagerie: système de transfert de messages: définition des services abstraits et procédures.*
- Recommandation UIT-T X.419 (1995) | ISO/CEI 10021-6:1996, *Technologies de l'information – Systèmes de messagerie: spécifications de protocole.*
- Recommandation UIT-T X.420¹⁾ | ISO/CEI 10021-7:...¹⁾, *Technologies de l'information – Systèmes de messagerie: système de messagerie de personne à personne.*

¹⁾ Actuellement à l'état de projet.

3 Définitions

3.1 Définitions communes pour le système de messagerie

On trouvera à la Rec. UIT-T X.402 | ISO/CEI 10021-2 une liste des définitions communes au système de messagerie.

3.2 Définitions concernant l'enregistrement des messages

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

3.2.1 association abstraite: Rattachement abstrait entre deux interlocuteurs; dans la présente Définition de service, rattachement entre un utilisateur MS et une mémoire de messages pour la fourniture du service abstrait mémoire de messages, ou rattachement entre une mémoire de messages et le système de transfert de messages (MTS) pour la fourniture du service abstrait système de transfert de messages.

3.2.2 accès d'administration: Accès prenant en charge les services de gestion (du système de transfert de messages) à l'intérieur du service abstrait de mémoire de messages.

3.2.3 opération abstraite avertissement: Opération abstraite permettant à la mémoire de messages (MS) de signaler à l'utilisateur MS qu'un message ou un rapport a été remis à la mémoire de messages. Cette opération ne peut être exécutée que sur une association abstraite existante.

3.2.4 attribut: Information d'un type particulier apparaissant dans une entrée.

3.2.5 type d'attribut: Composante d'un attribut indiquant la classe d'information donnée par cet attribut.

3.2.6 valeur d'attribut: Instance particulière de la classe d'information indiquée par un type d'attribut.

3.2.7 assertion de valeur d'attribut: Proposition *vraie, fausse* ou *indéfinie* concernant les valeurs d'un attribut d'une entrée.

3.2.8 action automatique: Actions qui sont exécutées automatiquement par la mémoire de messages (MS) selon des instructions préalablement enregistrées par l'utilisateur MS.

3.2.9 journal d'actions automatiques: Classe d'entrées contenant des entrées qui enregistrent l'exécution de certaines actions automatiques par la mémoire de messages.

3.2.10 événement d'action automatique: Entrée de la classe d'entrées Auto-action-log (journal d'action automatique) représentant l'exécution d'une action automatique.

3.2.11 type d'action automatique: Type d'une action automatique, par exemple Auto-alert (avertissement automatique).

3.2.12 avertissement automatique: Action automatique qui avertit l'utilisateur MS de la remise d'un message ou d'un rapport.

3.2.13 corrélation automatique des rapports: Action automatique qui établit une corrélation entre les rapports de remise et les messages ou messages d'essai déposés.

3.2.14 suppression automatique: Action automatique qui supprime les messages dont la période d'enregistrement est venue à expiration.

3.2.15 retransmission automatique: Classe d'actions automatiques qui entraîne la retransmission par la mémoire de messages d'un message remis à un ou plusieurs destinataires. Comme la définition de la retransmission automatique est propre au contenu, cette définition ne figure pas dans la présente Définition de service. Mais chaque type d'action de retransmission automatique est défini dans la Spécification pour le type de contenu concerné.

3.2.16 modification automatique: Action automatique qui applique les modifications aux attributs d'entrées nouvellement créées.

3.2.17 entrée fille: Entrée immédiatement subordonnée à une autre entrée (son entrée mère) dans une structure arborescente. Une entrée qui n'est pas une entrée fille est une entrée principale.

- 3.2.18 numéro de séquence d'une entrée fille:** Numéro de séquence situé au niveau d'une entrée mère et qui pointe vers une entrée fille. Dans une entrée mère, on trouvera une valeur de numéro de séquence pointant vers chacune de ses entrées filles.
- 3.2.19 ensemble de contraintes:** Ensemble d'objets informationnels servant à imposer des contraintes aux valeurs de composantes appartenant à un ensemble ou à une séquence; voir la Rec. UIT-T X.682 | ISO/CEI 8824-3.
- 3.2.20 propre au contenu:** Décrit une spécification ou une action dont l'effet dépend du type de contenu du message traité.
- 3.2.21 heure de création:** Attribut enregistrant la date et l'heure de création de l'entrée par la mémoire de messages.
- 3.2.22 opération abstraite suppression:** Opération abstraite utilisée pour supprimer une ou plusieurs entrées d'une classe d'entrées spécifiée.
- 3.2.23 types d'information codée remis:** Attribut multi-valué indiquant les types d'information codée présents dans le contenu d'un message remis.
- 3.2.24 message remis:** Entrée des classes d'entrée Delivery (remise) ou Delivery-log (journal de remise) représentant un message remis.
- 3.2.25 rapport remis:** Entrée des classes d'entrée Delivery (remise) ou Delivery-log (journal de remise) représentant un rapport remis.
- 3.2.26 remise:** Classe d'entrées contenant les entrées qui représentent les messages et les rapports remis par le système de transfert de messages (MTS) à la mémoire de messages (MS).
- 3.2.27 journal de remise:** Classe d'entrées contenant les entrées qui fournissent, à des fins de journalisation, une représentation restreinte des messages et des rapports remis par le système de transfert de messages (MTS) à la mémoire de messages (MS).
- 3.2.28 brouillon:** Classe d'entrées contenant les entrées draft-message (brouillons de message).
- 3.2.29 brouillon de message:** Entrée de la classe d'entrées Draft (brouillon) représentant un message non encore déposé auprès du système de transfert de messages (MTS).
- 3.2.30 entrée:** Objet informationnel stocké dans la mémoire de messages. On se reportera aux 3.2.17, 3.2.45 et 3.2.61 pour une classification plus poussée des entrées.
- 3.2.31 classe d'entrée:** Catégorie d'entrée représentant un type particulier d'objet informationnel. Les principales classes d'entrée sont les suivantes: Stored-message (message enregistré), Message-log (journal de messagerie) et Auto-action-log (journal d'actions automatiques).
- 3.2.32 information d'entrée:** Paramètre utilisé dans les opérations abstraites pour transmettre une information sélectionnée à partir d'une entrée.
- 3.2.33 sélection d'information d'entrée:** Paramètre utilisé dans les opérations abstraites pour indiquer quelle information d'une entrée est demandée.
- 3.2.34 type d'entrée:** Attribut indiquant si une entrée est de type delivered-message (message remis), delivered-report (rapport remis), returned-content (contenu renvoyé), submitted-message (message déposé), submitted-probe (message d'essai déposé), draft-message (brouillon de message) ou auto-action-event (événement d'action automatique).
- 3.2.35 opération abstraite recherche:** Opération abstraite qui permet à un utilisateur MS de rechercher à partir de la mémoire de messages un ensemble non restreint d'informations relatives aux attributs d'une entrée sélectionnée d'une classe d'entrées spécifiée.
- 3.2.36 restrictions de recherche:** Restrictions imposées par l'utilisateur MS sur le type d'information qu'il est prêt à recevoir à la suite d'une recherche. Les restrictions peuvent porter sur la longueur de l'attribut, le type de contenu et le type d'information codée.
- 3.2.37 filtre:** Paramètre utilisé dans les opérations abstraites qui consiste à appliquer un test à une entrée, le test étant ensuite vérifié ou non par cette entrée.
- 3.2.38 élément de filtre:** Assertion sur la présence ou sur la ou les valeurs d'un attribut de type particulier dans une entrée sous test. Cette assertion peut être *vraie*, *fausse* ou *indéfinie*.

- 3.2.39 demande de retransmission:** Paramètre qui peut être présent dans l'argument de l'opération abstraite MS-message-submission (dépôt de messages MS) invoquée par l'utilisateur MS pour demander à la mémoire de messages de retransmettre un message enregistré.
- 3.2.40 attribut général:** Attribut valable pour tous les types de messages et de rapports, quel que soit le type de contenu. Seuls les attributs de ce type sont définis dans la présente Définition de service.
- 3.2.41 action automatique générale:** Action automatique valable pour tous les types de messages et de rapports, quel que soit le type de contenu. Seules les actions automatiques de ce type sont définies dans la présente Définition de service.
- 3.2.42 catégorie:** Défini au 5.2 de la Rec. UIT-T X.402 | ISO/CEI 10021-2.
- 3.2.43 limite:** Composante du paramètre selector (sélecteur) spécifiant le nombre maximal d'entrées sélectionnées à renvoyer comme résultat d'une opération abstraite.
- 3.2.44 opération abstraite listage:** Opération abstraite permettant de sélectionner les entrées d'une classe d'entrées spécifiée et demandant de renvoyer des informations relatives à certains attributs de ces entrées.
- 3.2.45 entrée principale:** Entrée pouvant former la racine d'un ensemble d'entrées reliées, organisées en structure arborescente. Zéro, une ou plusieurs entrées filles peuvent être associées à une entrée principale.
- 3.2.46 règle de criblage:** Règle permettant de sélectionner des entrées en énonçant des assertions concernant les valeurs de leurs attributs.
- 3.2.47 groupe de messages:** Ensemble d'entrées reliées. L'attribut d'entrée message-group-name (nom du groupe de messages) indique le groupe de messages auxquels l'ensemble appartient.
- 3.2.48 journal de messagerie:** Classe d'entrées qui comprend toutes les entrées des classes d'entrée Submission-log (journal de dépôt) et Delivery-log (journal de remise).
- 3.2.49 opération abstraite dépôt de messages:** Opération abstraite que la mémoire de messages invoque pour déposer un message auprès du système de transfert de messages lorsque l'utilisateur MS invoque l'opération abstraite de dépôt MS-message-submission ou lorsque certaines actions automatiques propres au contenu sont exécutées.
- 3.2.50 opération abstraite modification:** Opération abstraite servant à modifier les attributs d'une ou plusieurs entrées à l'intérieur d'une classe d'entrée.
- 3.2.51 service abstrait de mémoire de messages:** Ensemble des capacités que la mémoire de messages offre à ses utilisateurs à travers ses accès.
- 3.2.52 utilisateur du service abstrait MS:** Utilisateur du service abstrait de mémoire de messages. Un utilisateur final peut utiliser différents agents utilisateur (UA), à des instants différents, pour agir en tant qu'utilisateur du service abstrait de mémoire de messages lorsqu'il accède à ce service.
- 3.2.53 fournisseur de service abstrait de mémoire de messages:** Mémoire de messages fournissant le service abstrait de mémoire de messages.
- 3.2.54 opération abstraite dépôt de messages MS:** Opération abstraite qui permet à l'utilisateur MS de déposer un message auprès du système de transfert de messages (et facultativement d'en enregistrer une copie), ou d'enregistrer un brouillon de message.
- 3.2.55 opération abstraite dépôt de message d'essai MS:** Opération abstraite qui permet à l'utilisateur MS de déposer un message d'essai auprès du système de transfert de messages, et facultativement d'en enregistrer une copie.

- 3.2.56 accès de dépôt MS:** Accès qui prend en charge les services de dépôt MS du service abstrait de mémoire de messages. Le service abstrait de dépôt MS-submission offre les mêmes services que le service abstrait de dépôt Message-submission (du service abstrait de système de transfert de messages), plus des services d'enregistrement et de journalisation des messages déposés, et de retransmission de messages se trouvant dans la mémoire de messages.
- 3.2.57 options de dépôt MS:** Paramètre utilisé dans les opérations abstraites et dans les actions automatiques pour déterminer si un message ou un message d'essai doit être déposé auprès du système de transfert de messages, ou enregistré dans la mémoire de messages ou à la fois déposé et enregistré.
- 3.2.58 utilisateur MS:** Forme abrégée de l'expression «utilisateur du service abstrait de mémoire de messages».
- 3.2.59 attribut multi-valué:** Attribut auquel peuvent être associées plusieurs valeurs.
- 3.2.60 ignorer:** Composante du paramètre selector (sélecteur) indiquant que les restrictions précédemment enregistrées pour l'exécution de l'opération abstraite Fetch (recherche) ne doivent pas s'appliquer à cette instance de l'opération abstraite.
- 3.2.61 entrée mère:** Entrée immédiatement supérieure à une ou plusieurs entrées filles. Une entrée mère qui n'est pas l'entrée fille d'une autre entrée est une entrée principale.
- 3.2.62 numéro de séquence d'entrée mère:** Numéro de séquence qui identifie l'entrée mère d'une entrée fille. Chaque entrée fille possède une valeur de numéro de séquence d'entrée mère.
- 3.2.63 demande d'attribut partiel:** Forme du paramètre entry-information-selection (sélection d'informations d'entrée) demandant de renvoyer uniquement les valeurs sélectionnées d'un attribut multi-valué; voir 3.2.33.
- 3.2.64 intervalle:** Paramètre utilisé dans les opérations abstraites pour sélectionner une séquence contiguë d'entrées d'une classe d'entrées spécifiée.
- 3.2.65 opération abstraite consignation MS:** Opération abstraite permettant à l'utilisateur MS de consigner des informations concernant son interfonctionnement avec la mémoire de messages.
- 3.2.66 consignation:** Information consignée par la mémoire de messages et conservée entre deux établissements d'association abstraite [jusqu'à ce qu'elle soit modifiée par une opération abstraite Register-MS (consignation MS)]; voir 3.2.65.
- 3.2.67 identificateur de consignation:** Identificateur pour un ensemble particulier de paramètres de consignation pour un type d'action automatique.
- 3.2.68 accès d'extraction:** Accès prenant en charge les services d'extraction du service abstrait MS.
- 3.2.69 état de l'extraction:** Attribut qui indique si une entrée a été soumise à une extraction par l'utilisateur MS dans la mémoire de messages. Les valeurs possibles sont: *new* (nouvelle), *listed* (listée) et *processed* (traitée).
- 3.2.70 contenu renvoyé:** Entrée de la classe d'entrées Delivery (remise) dans laquelle figure le contenu renvoyé d'un message précédemment déposé.
- 3.2.71 sélecteur:** Paramètre utilisé dans les opérations abstraites pour sélectionner les entrées d'une classe d'entrées spécifiée.
- 3.2.72 numéro de séquence:** Attribut identifiant sans ambiguïté une entrée. Les numéros de séquence sont attribués par ordre croissant.
- 3.2.73 attribut mono-valué:** Attribut auquel ne peut être associée qu'une seule valeur.
- 3.2.74 message enregistré:** Classe d'entrées comprenant toutes les entrées des classes d'entrée Submission (dépôt) et Delivery (remise).
- 3.2.75 dépôt:** Classe d'entrées comprenant les entrées qui représentent les messages et les messages d'essai qui ont été déposés auprès du système de transfert de messages via la mémoire de messages.

3.2.76 journal de dépôt: Classe d'entrées comprenant les entrées qui fournissent, à des fins de journalisation, une représentation restreinte des messages et des messages d'essai qui ont été déposés auprès du système de transfert de messages.

3.2.77 message déposé: Entrée des classes d'entrée Submission (dépôt) ou Submission-log (journal de dépôt) représentant un message déposé.

3.2.78 message d'essai déposé: Entrée des classes d'entrée Submission (dépôt) ou Submission-log (journal de dépôt) représentant un message d'essai déposé.

3.2.79 abonnement: Accord à long terme entre le fournisseur ou le gérant du service MS et les utilisateurs de ce service sur la mise à disposition et l'utilisation des fonctionnalités facultatives de la mémoire de messages. Dans la présente Définition de service, on suppose qu'un tel accord peut être conclu sans toutefois prescrire la méthode qui permet de le conclure.

3.2.80 opération abstraite récapitulation: Opération abstraite qui fournit un aperçu du genre et du nombre d'entrées appartenant à une certaine classe qui sont enregistrées à ce moment-là dans la mémoire de messages.

3.2.81 consignment d'agent utilisateur: Information consignée par la mémoire de messages propre à l'un des agents utilisateur (UA) de l'utilisateur; voir 3.2.66.

4 Abréviations

La présente Définition de service utilise les abréviations définies à l'article 5 de la Rec. UIT-T X.402 | ISO/CEI 10021-2, plus celles qui sont indiquées ci-dessous.

ASN.1	Notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
DL	Liste de distribution (<i>distribution-list</i>)
EIT	Type d'information codée (<i>encoded-information-type</i>)
MASE-88	Élément de service de gestion de messages, version 1988 (<i>message administration service element 1988</i>)
MASE-94	Élément de service de gestion de messages, version 1994 (<i>message administration service element 1994</i>)
MHS	Systèmes de messagerie (<i>message handling systems</i>)
MRSE-88	Élément de service d'extraction de message, version 1988 (<i>message retrieval service element 1988</i>)
MRSE-94	Élément de service d'extraction de message, version 1994 (<i>message retrieval service element 1994</i>)
MS	Mémoire de messages; service de mémoire de messages (service MS) (<i>message store</i>)
MSSE	Élément de service de dépôt de messages (<i>message submission service element</i>)
MS-MSSE	Élément de service de dépôt de messages MS (<i>MS message submission service element</i>)
MTS	Système de transfert de messages (<i>message transfer system</i>)
ROS	Service d'opérations distantes (<i>remote operation service</i>)
UA	Agent utilisateur (<i>user agent</i>)
UTC	Temps universel coordonné (<i>universal coordinated time</i>) (voir la Rec. UIT-T X.680 ISO/CEI 8824-1)

5 Conventions

La présente Définition de service utilise les conventions descriptives énumérées ci-dessous.

5.1 Conventions applicables aux services abstraits

La présente Définition de service utilise les conventions descriptives suivantes de l'ASN.1 pour définir les entités ci-dessous:

- a) *objets informationnels du service abstrait de mémoire de messages et autres valeurs et types de données*: la notation ASN.1;
- b) *service abstrait de mémoire de messages*: les classes d'objets informationnels MHS-OBJECT (objet du système de messagerie), PORT (accès), ABSTRACT-OPERATION (opération abstraite) et ABSTRACT-ERROR (erreur abstraite) de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Celles-ci dérivent directement des classes d'objets informationnels correspondantes définies dans la Rec. UIT-T X.880 | ISO/CEI 13712-1, qui définit également les classes d'objets informationnels CONTRACT (contrat) et CONNECTION-PACKAGE (bloc de connexions);
- c) *attributs*: la classe d'objets informationnels ATTRIBUTE (attribut) du 6.3.3.3;
- d) *actions automatiques*: les classes d'objets informationnels AUTO-ACTION (action automatique) et AUTO-ACTION-ERROR (erreur d'action automatique) du 6.5;
- e) *règles de criblage*: la classe d'objets informationnels MATCHING-RULE (règle de criblage) de la Rec. UIT-T X.501 | ISO/CEI 9594-2; voir également 6.3.9.3.

Chaque fois que la présente Définition de service décrit une classe de données structurées comportant un certain nombre de composantes, chaque composante sera classée dans l'une des **catégories** ci-après:

- a) **obligatoire (M)**: une composante obligatoire doit figurer dans chaque instance de la classe;
- b) **optionnel (O)**: une composante optionnelle peut figurer dans une instance de la classe à la discrétion de l'objet (par exemple l'utilisateur) fournissant cette instance;
- c) **conditionnel (C)**: une composante conditionnelle figurera ou non dans une instance de la classe selon la spécification de la présente Définition de service.

5.2 Conventions utilisées dans les Tableaux 2 et 3 de l'article 11 et applicables aux types d'attributs

La présente Définition de service utilise les conventions énumérées ci-après pour définir les types d'attributs utilisés dans le service abstrait MS.

Pour la colonne intitulée «mono-/multi-valué» on peut avoir les valeurs suivantes:

- S mono-valué
- M multi-valué

Dans la colonne intitulée «Prise en charge par la mémoire de messages» du Tableau 2, le sous-titre «Classe d'entrées Stored-message (message enregistré)» fait référence aux classes d'entrée Stored-message [et aux classes d'entrée Submission (dépôt), Delivery (remise) et Draft (brouillon)], et le sous-titre «Classe d'entrées Message-log (journal de messagerie)» fait référence aux classes d'entrée Message-log [et aux classes d'entrée Submission-log (journal de dépôt) et Delivery-log (journal de remise)]. Les valeurs suivantes peuvent figurer dans cette colonne des Tableaux 2 et 3:

- M obligatoire
- O optionnel
- non pris en charge

Dans les colonnes intitulées «Présence dans l'entrée» delivered-message (message remis), delivered-report (rapport remis), returned-message (message renvoyé), submitted-message (message déposé), submitted-probe (message d'essai déposé), draft-message (brouillon de message) et de la classe d'entrées Auto-action-log (journal d'actions automatiques), la présence de chaque type d'attribut est caractérisée par l'une des valeurs suivantes:

- P Toujours présent dans l'entrée parce que:
 - cet attribut est obligatoire aux fins de création par la mémoire de messages;
 - cet attribut est un paramètre obligatoire ou attribué par défaut dans l'opération abstraite correspondante.

- C Présent de façon conditionnelle dans l'entrée. Ce paramètre est présent si:
- il est fourni par la mémoire de messages et fait l'objet d'un abonnement de la part de l'utilisateur;
 - il figure dans un paramètre optionnel de l'opération abstraite correspondante.
 - Toujours absent dans les autres cas.

Dans les colonnes intitulées «Disponible pour List (listage)» et «Disponible pour Summarize (récapitulation)», on peut avoir les valeurs suivantes:

- N non
Y oui

5.3 Conventions applicables aux types d'attributs utilisés dans le Tableau 4 de l'article 11

La présente Définition de service utilise les conventions ci-dessous pour définir les types d'attributs applicables au service abstrait de mémoire de messages.

Dans la colonne intitulée «mono-valué/multi-valué», on peut avoir les valeurs suivantes:

- S mono-valué
M multi-valué

Dans la colonne intitulée «Source créée par», on peut avoir les valeurs suivantes:

- Amod Action automatique Auto-modify (modification automatique)
Md Opération abstraite Message-delivery (remise de messages)
Mod Opération abstraite Modify (modification)
MS Mémoire de messages; service de mémoire de messages (service MS)
Ms Opérations abstraites Message-submission (dépôt de message) et MS-message-submission (dépôt de message MS)
Ps Opérations abstraites Probe-submission (dépôt de message d'essai) et MS-probe-submission (dépôt de message d'essai MS)
Rd Opération abstraite Report-delivery (remise de rapport)

5.4 Conventions générales d'écriture

Dans la présente Définition de service, les termes définis sont **écrits en caractères gras**. Les noms propres sont écrits en majuscules, et les noms génériques en minuscules. Les noms génériques à plusieurs termes sont reliés par des traits d'union.

5.5 Conventions d'écriture applicables aux définitions ASN.1

Dans la présente Définition de service, les définitions ASN.1 sont écrites en caractères Courier pour les distinguer du texte normal. Le corps des caractères ASN.1 sera d'un point plus petit que celui du texte normal.

5.6 Règles applicables aux définitions ASN.1

Les définitions ASN.1 apparaissent à la fois dans le corps de la présente Définition de service pour faciliter la présentation et, de manière formelle, dans une annexe à titre de référence. Si des différences apparaissent entre les déclarations ASN.1 du corps du document et les définitions formelles de l'annexe correspondante, une erreur de spécification est signalée.

5.7 Conventions applicables aux éditions antérieures de la présente Définition de service

La présente Définition de service utilise l'expression «contextes d'application 1988» pour faire référence au fonctionnement du service de mémoire de messages avec les contextes d'application ms-access-88 (accès MS) et ms-reliable-access-88 (accès fiable MS) définis dans la Rec. X.419 du CCITT (1988) et (1992) | ISO/CEI 10021-6:1990. L'expression «contextes d'application 1994» fait référence au fonctionnement du service de mémoire de messages avec les contextes d'application ms-access-94 et ms-reliable-access-94 définis dans la Rec. UIT-T X.419 (1995) | ISO/CEI 10021-6:1990, Amendement 1:1994.

On peut faire correspondre la syntaxe abstraite définie dans la présente Définition de service avec celle utilisée dans les précédentes éditions de la façon suivante. Toutes les définitions ASN.1 d'ensembles d'objets et de types énumérés qui contiennent le marqueur d'extensions ASN.1 («...») sont traitées comme si toutes les extensions supplémentaires qui suivent le marqueur étaient absentes. Pour les définitions n'utilisant pas le marqueur d'extension, le commentaire ASN.1 «-- extension 1994 --» est interprété de la même façon. Ceci a pour conséquence que l'utilisation de certaines opérations abstraites, classes d'entrée, types d'entrée, types d'attributs, règles de criblage et types d'actions automatiques n'est pas normalisée dans les contextes d'application 1988. De plus, les types d'attributs et les colonnes repérés par un astérisque (*) dans le Tableau 2 ne sont pas définis pour les contextes d'application 1988.

SECTION 2 – DÉFINITION DU SERVICE ABSTRAIT D'ENREGISTREMENT DE MESSAGES

6 Modèle de mémoire de messages (MS)

La mémoire de messages est modélisée par un objet qui joue le rôle de fournisseur de services destinés à l'utilisateur du service abstrait de mémoire de messages (l'utilisateur MS) et d'utilisateur des services fournis par le système de transfert de messages (MTS).

La mémoire de messages joue un rôle d'intermédiaire entre l'utilisateur MS et le système de transfert de messages (MTS). Sa fonction principale est d'assurer l'enregistrement des messages déposés par ou remis à un utilisateur final unique d'un système de messagerie, et de les conserver pour que le ou les agents utilisateur (UA) de l'utilisateur final puissent par la suite les extraire. La mémoire de messages fournit également à l'utilisateur MS des services de dépôt de messages et de gestion de messages en relais vers le système de transfert de messages (MTS). La mémoire de messages offre ainsi des fonctionnalités nouvelles par rapport au dépôt direct de messages auprès du système de transfert de messages, par exemple l'enregistrement des messages déposés, la retransmission de messages contenus dans la mémoire de messages et la journalisation d'événements de dépôt et de remise.

Comme l'utilisateur MS, la mémoire de messages agit pour le compte d'un utilisateur final unique du système de messagerie, c'est-à-dire qu'elle n'assure pas un service de mémoire de messages multi-utilisateurs commun ou partagé.

NOTE – Dans une réalisation type, un système unique desservira plusieurs utilisateurs et comprendra donc plusieurs mémoires de messages.

La mémoire de messages est décrite à l'aide d'un modèle abstrait pour définir le service qu'elle assure – le service abstrait de mémoire de messages. La Figure 1 représente le service abstrait de mémoire de messages et ses liens avec l'utilisateur MS et le service abstrait de transfert de messages (MTS). Dans cette figure, les carrés blancs représentent l'utilisation d'un service abstrait, et les carrés noirs la fourniture d'un service abstrait.

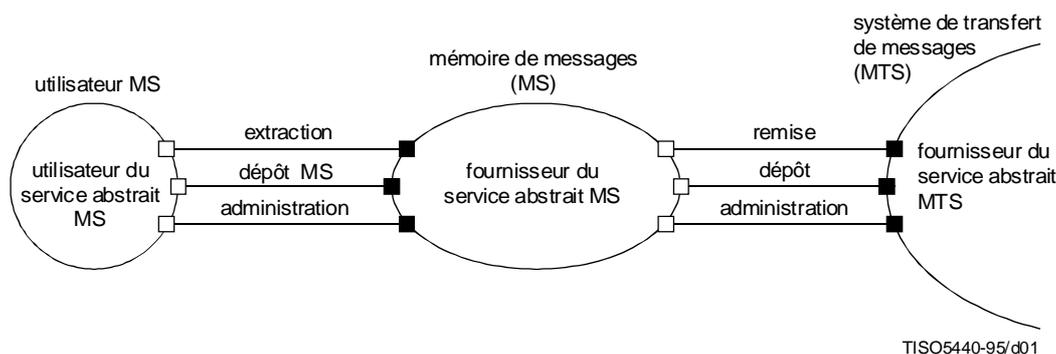


Figure 1 – Service abstrait d'enregistrement des messages

On trouvera à la Rec. UIT-T X.402 | ISO/CEI 10021-2 une introduction et une description de la notion de service abstrait et des conventions de définition qui lui sont associées. Ces conventions utilisent diverses classes d'objets informationnels utilisées pour la spécification des protocoles d'application basés sur le service d'opérations distantes (ROS) qui sont définis dans la Rec. UIT-T X.880 | ISO/CEI 13712-1.

Dans le cadre de la messagerie sécurisée, la mémoire de messages est traitée comme un objet distinct de l'utilisateur MS, possédant une identité unique et une clé ou un ensemble de clés distincts.

6.1 Contrats et objets de la mémoire de messages

La **mémoire de messages** est modélisée par un objet qui fournit le service abstrait de mémoire de messages à l'utilisateur MS selon les termes d'un contrat d'accès MS. La mémoire de messages est le répondeur des contrats d'accès MS-access-contract-88 et MS-access-contract-94, respectivement associés aux contextes d'application 1988 et 1994.

```
ms MHS-OBJECT ::= {
  IS      {mts-user}
  RESPONDS {ms-access-contract-88 | ms-access-contract-94}
  ID      id-ot-ms }
```

Pour assurer l'accès de l'utilisateur MS aux services du système de transfert de messages, la mémoire de messages joue le rôle d'un objet utilisateur MTS: voir la Figure 2 (partie 2) et l'article 8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Dans ce rôle, elle utilise les services abstraits de dépôt, de remise et de gestion du système de transfert de messages et joue le rôle de fournisseur de services abstraits de gestion et de dépôt MTS pour l'utilisateur MS.

L'**utilisateur MS** est également modélisé par un objet. Il lance les contrats d'accès MS-access-contract-94 et MS-access-contract-88 en vertu desquels il joue le rôle de client du service abstrait de mémoire de messages.

```
ms-user MHS-OBJECT ::= {
  INITIATES {ms-access-contract-88 | ms-access-contract-94}
  ID      id-ot-ms-user }
```

Le contrat d'accès **MS-access-contract-94** définit un contrat d'association établi au moyen du bloc de connexion MS-connect, en vertu duquel le demandeur (l'utilisateur MS) joue le rôle de client des services abstraits d'accès d'extraction et d'accès de dépôt MS fournis directement par la mémoire de messages, et des services abstraits d'accès d'administration fournis de façon transparente par la mémoire de messages.

```
ms-access-contract-94 CONTRACT ::= {
  CONNECTION      ms-connect
  INITIATOR CONSUMER OF {retrieval | ms-submission | administration}
  ID      id-crt-ms-access-94 }
```

Le contrat d'accès **MS-access-contract-88** définit un contrat d'association établi au moyen du bloc de connexion MS-connect, en vertu duquel le demandeur (l'utilisateur MS) joue le rôle de client des services abstraits d'accès d'extraction 88 fournis directement par la mémoire de messages, et des services abstraits de dépôt et d'accès d'administration 88 fournis en transparence par la mémoire de messages.

```
ms-access-contract-88 CONTRACT ::= {
  CONNECTION      ms-connect -- toutes les additions d'extension 1994 sont omises --
  INITIATOR CONSUMER OF {retrieval-88 | submission | administration-88}
  ID      id-crt-ms-access-88 }
```

Le bloc de connexion **MS-connect** spécifie les opérations utilisées pour établir et libérer une association abstraite entre la mémoire de messages et l'utilisateur MS.

```
ms-connect CONNECTION-PACKAGE ::= {
  BIND      ms-bind
  UNBIND    ms-unbind
  ID      id-cp-ms-connection }
```

6.2 Accès de la mémoire de messages

La mémoire de messages fournit à l'utilisateur MS les accès d'extraction, de dépôt MS et d'administration. L'ensemble des capacités offertes par ces accès constitue le service abstrait de mémoire de messages (MS). Les capacités d'extraction de messages ne sont offertes que par la mémoire de messages. Ces capacités comprennent la récapitulation des messages, la recherche de messages (en totalité ou en partie), la suppression de messages contenus dans la mémoire de messages et la consignation des requêtes adressées à la mémoire de messages demandant de réaliser automatiquement certaines activités de gestion des messages.

La mémoire de messages assure des services supplémentaires de gestion des messages pour le compte de l'utilisateur MS, comme la journalisation des messages entrants et sortants, la classification automatique des messages enregistrés et la corrélation automatique des rapports de remise avec les messages ou les messages d'essai déposés auxquels ils se rapportent.

A l'aide de l'opération abstraite MS-bind (rattachement MS), la mémoire de messages authentifie l'utilisateur MS avant de mettre à sa disposition les capacités d'extraction et d'enregistrement décrites ci-dessus. De même, le service abstrait MTS authentifiera l'utilisateur du service abstrait MTS avant d'offrir ses services à cet utilisateur (c'est-à-dire la mémoire de messages).

A l'exception du service Alert (avertissement) de l'accès d'extraction, du service Submission-control (commande de dépôt) de l'accès de dépôt MS et du service Change-credentials (modification des pouvoirs) de l'accès d'administration, tous les services fournis par le service abstrait de mémoire de messages sont invoqués par l'utilisateur MS et exécutés par la mémoire de messages.

Des étiquettes de sécurité peuvent être affectées à la mémoire de messages conformément à la politique de sécurité en vigueur. Cette politique peut également définir ses propres modalités d'utilisation des étiquettes de sécurité. Si des étiquettes de sécurité sont affectées à la mémoire de messages, le traitement des messages et des rapports enregistrés portant une étiquette de sécurité de message peut être affecté par la politique de sécurité en vigueur. Si aucune étiquette de sécurité n'est affectée à la mémoire de messages, le traitement des messages et des rapports enregistrés comportant des étiquettes de sécurité de message est du ressort local.

Si des contextes de sécurité sont établis entre l'agent utilisateur (UA) et la mémoire de messages, ainsi qu'entre la mémoire de messages et le système de transfert de messages (MTS), l'étiquette de sécurité affectée à un message ou à un message d'essai est limitée au contexte de sécurité conformément à la politique de sécurité en vigueur. En l'absence d'un contexte de sécurité, l'affectation d'une étiquette de sécurité à un message ou à un message d'essai est au choix de l'expéditeur.

6.2.1 Accès d'extraction

L'accès d'extraction **Retrieval Port** est défini comme suit:

```
retrieval PORT ::= {
  CONSUMER INVOKES {summarize | list | fetch | delete | register-MS,
    ... -- extension de 1994 --,
    modify}
  SUPPLIER INVOKES {alert}
  ID id-pt-retrieval-94 }
```

Les services abstraits de l'accès d'extraction **Retrieval Port** sont décrits en détail au 8.2 et à l'article 9.

L'accès d'extraction **Retrieval-88 Port** fourni selon les termes du contrat d'accès MS-access-contract-88 est défini comme suit:

```
retrieval-88 PORT ::= {
  -- aucune des extensions de 1994 relatives aux opérations abstraites ne figure --
  CONSUMER INVOKES {summarize | list | fetch | delete | register-MS}
  SUPPLIER INVOKES {alert}
  ID id-pt-retrieval-88 }
```

6.2.2 Accès de dépôt MS

L'accès de dépôt MS **MS-submission Port** est défini comme suit:

```
ms-submission PORT ::= {
  CONSUMER INVOKES {ms-message-submission | ms-probe-submission | ms-cancel-deferred-delivery}
  SUPPLIER INVOKES {ms-submission-control}
  ID id-pt-ms-submission }
```

Les services abstraits de l'accès de dépôt **MS-submission Port** sont décrits en détail au 8.3 et à l'article 9.

L'accès de dépôt **Submission Port** fourni selon les termes du contrat d'accès MS-access-contract-88 est défini au 8.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

6.2.3 Accès d'administration

L'accès d'administration **Administration Port** est défini au 8.4 (et l'accès d'administration **Administration-88 Port** dans l'Annexe C) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

Lorsqu'il est demandé par l'utilisateur MS, le service abstrait de modification des pouvoirs fonctionne de bout en bout entre l'utilisateur MS et le fournisseur du service du système de transfert de messages via la mémoire de messages. Celle-ci enregistre les nouveaux pouvoirs à utiliser lors de ses prochains rattachements au système MTS. Si l'utilisateur MS a besoin de mettre à jour les pouvoirs qu'il utilise pour se rattacher à la mémoire de messages, il utilisera l'opération abstraite Register-MS (consignation MS); voir 8.2.5.

6.3 Modèle informationnel

La mémoire de messages modélise chaque objet informationnel qu'elle enregistre comme une **entrée**, qui se compose d'un ensemble d'informations connues sur cet objet. Chaque entrée appartient à une classe d'entrées **entry-class**, qui identifie la classe d'objets décrite par l'entrée. Une entrée comprend un ensemble d'**attributs**, qui représentent les éléments composant les informations associées à l'entrée. Les types d'attribut présents dans une entrée dépendent, en partie, de la classe de l'entrée.

6.3.1 Classes d'entrée

Une classe d'entrées **entry-class** comprend une ou plusieurs catégories d'entrées qui représentent un type donné d'objet informationnel. On définit trois classes d'entrée principales: la classe d'entrées **Stored-message** (message enregistré), la classe d'entrées **Message-log** (journal de messagerie) et la classe d'entrées **Auto-action-log** (journal d'actions automatiques).

- a) La classe d'entrées **Stored-message** (message enregistré) contient les entrées qui correspondent aux messages, rapports, messages d'essai et brouillons de messages.
- b) La classe d'entrées **Message-log** (journal de messagerie) contient les entrées qui donnent une représentation restreinte des messages, rapports et messages d'essai. Ces entrées ressemblent étroitement aux entrées correspondantes de la classe d'entrées **Stored-message** (message enregistré), mais, conformément à leur fonction de journalisation, ne contiennent qu'un sous-ensemble des attributs présents dans cette dernière.
- c) La classe d'entrées **Auto-action-log** (journal d'actions automatiques) contient les entrées qui enregistrent l'exécution des actions automatiques par la mémoire de messages; voir 6.3.7.3.

On définit cinq classes d'entrée subordonnées qui contiennent des sous-ensembles des entrées contenues dans les classes d'entrées principales. Les classes d'entrées **Delivery** (remise), **Submission** (dépôt) et **Draft** (brouillon) sont subordonnées à la classe d'entrées **Stored-message** (message enregistré). Les classes d'entrées **Delivery-log** (journal de remise) et **Submission-log** (journal de dépôt) sont subordonnées à la classe d'entrées **Message-log** (journal de messagerie). Le paragraphe 6.3.7 donne de plus amples détails sur l'organisation des classes d'entrée.

Lorsque l'utilisateur MS invoque une opération abstraite du service abstrait de mémoire de messages, il en limite l'effet aux entrées immédiatement concernées en spécifiant la classe des entrées sur lesquelles il faut opérer.

```
EntryClass ::= INTEGER {
    delivery                (0),
    -- extensions de 1994 --
    submission              (1),
    draft                   (2),
    stored-message          (3),
    delivery-log            (4),
    submission-log          (5),
    message-log             (6),
    auto-action-log         (7) } (0..ub-entry-classes)
```

NOTES

1 L'expression *base d'information* utilisée dans la Rec. X.413 du CCITT (1988) et (1992) | ISO/CEI 10021-5:1990 a été remplacée par *classe d'entrées* dans la présente Définition de service.

2 Dans la présente Définition de service, la description du service abstrait de mémoire de messages suppose que toutes les classes d'entrée définies sont disponibles et peuvent être utilisées. En pratique, le comportement d'une réalisation donnée de mémoire de messages dépendra de sa prise en charge de composantes optionnelles du service (par exemple les classes d'entrée et les types d'attribut optionnels) et de l'abonnement; voir 6.3.7.4.

6.3.2 Entrées

Chaque objet informationnel stocké dans la mémoire de messages (un message déposé ou un rapport remis par exemple) est représenté comme une **entrée**. L'entrée est identifiée par son numéro de séquence **sequence-number** généré par la mémoire de messages lors de la création de l'entrée. Les numéros de séquence, générés par ordre croissant, ne sont jamais réutilisés.

SequenceNumber ::= INTEGER (0..ub-messages)

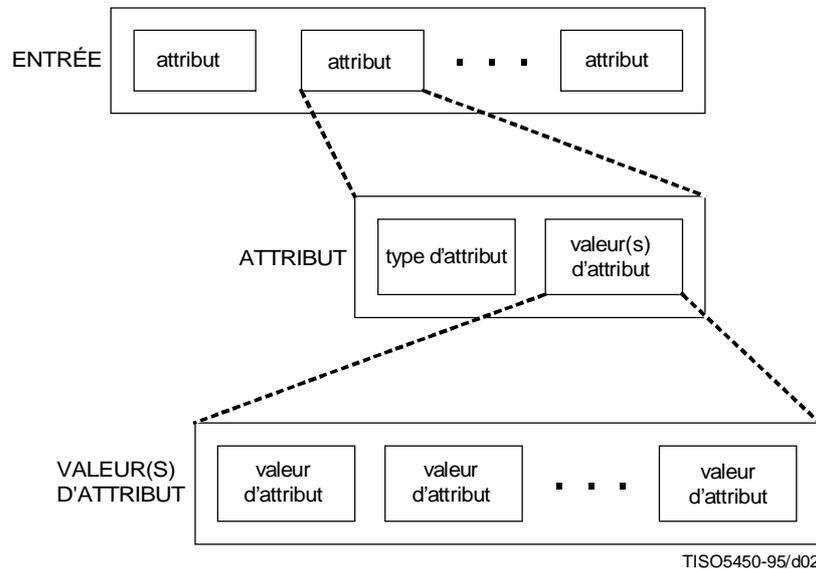
Un même numéro de séquence n'est jamais affecté à deux entrées de la mémoire de messages sauf dans le cas suivant. En raison de la relation étroite qui existe entre les entrées de la classe Stored-message (message enregistré) et celles de la classe Message-log (journal de messagerie), les entrées correspondantes de ces classes recevront les mêmes numéros de séquence.

NOTE – La mémoire de messages peut par exemple attribuer des numéros de séquence en utilisant l'heure de création de l'entrée avec une granularité suffisamment fine pour garantir l'unicité de numérotation.

Toutes les entrées ont les trois propriétés suivantes en commun. Le **type d'entrée** indique le type d'objet informationnel représenté par l'entrée, l'heure de création **creation-time** indique la date et l'heure de création de l'entrée et l'état d'extraction **retrieval-status** indique si l'entrée a été extraite par l'utilisateur MS.

6.3.3 Attributs

Une **entrée** est composée d'un ensemble d'attributs **attribut** comme le montre la Figure 2.



TISO5450-95/d02

Figure 2 – Composantes d'une entrée

Chaque attribut **attribut** donne un élément d'information sur les données ou dérivé des données auxquelles correspond l'entrée. A titre d'exemple de ce type d'information, on peut citer le numéro de séquence de l'entrée proprement dite et l'heure de création.

Chaque attribut est défini comme une instance de la classe d'objets informationnels **ATTRIBUTE**; voir 6.3.3.3.

Un attribut comprend un type **attribute-type** identifiant la classe de l'information contenue dans l'attribut, et une ou plusieurs valeurs **attribute-values** qui sont des instances particulières de cette classe figurant dans l'entrée. L'ensemble des contraintes Attribute-table (tableau des attributs) est défini au 11.2.

```
Attribute ::= SEQUENCE {
  attribute-type    ATTRIBUTE.&id ({AttributeTable}),
  attribute-values  SEQUENCE SIZE (1..ub-attribute-values) OF ATTRIBUTE.&Type
                    ({AttributeTable} {@attribute-type}) }
```

NOTE 1 – Ainsi, par exemple, dans une entrée delivered-message (message remis) (décrite au 6.3.6), le type d'attribut pourrait être la priorité du message et la valeur d'attribut correspondante pourrait être *urgent*.

Les types des attributs d'une entrée seront tous distincts.

Certains types d'attribut admettent une seule valeur; ils sont dits **mono-valués**. D'autres attributs admettent une ou plusieurs valeurs, appartenant toutes au même type de données ASN.1; ils sont dits **multi-valués**. Le type d'attribut est déclaré mono-valué ou multi-valué au moment de sa définition (voir 6.3.3.3).

NOTE 2 – Ainsi, par exemple, le type d'attribut originator-name (nom de l'expéditeur) (décrit au 11.2.47) est mono-valué, alors que le type d'attribut other-recipients-names (noms des autres destinataires) (décrit au 11.2.50) est multi-valué.

Afin de sélectionner les entrées de la mémoire de messages selon les valeurs de leurs attributs, un ensemble de règles de criblage peut être associé au type d'attribut pour spécifier le type de cribles qui peuvent s'appliquer à ce type d'attribut (voir 6.3.9).

6.3.3.1 Type d'attribut

Certains types d'attribut sont normalisés au niveau international. D'autres peuvent être définis par les autorités administratives nationales et par des organisations privées. C'est-à-dire qu'un certain nombre d'autorités distinctes seront responsables de l'affectation de types d'attribut et qu'elles devront s'assurer que ce type est distinct de tous les autres types affectés par ailleurs. A cet effet, chaque type d'attribut est identifié au moment de sa création par un identificateur d'objet Object Identifier.

```
AttributeType ::= OBJECT IDENTIFIER
```

Certains types d'attribut de caractère général applicables aux entrées de tous les types de contenu sont définis à l'article 11. Ces types d'attribut sont appelés **types d'attribut général** et les attributs correspondants des **attributs généraux**.

6.3.3.2 Valeurs d'attributs

Définir un type d'attribut suppose également qu'on en spécifie la syntaxe et donc le type de données auquel chaque valeur de l'attribut doit être conforme. Le type de données d'une valeur d'attribut pour le type d'attribut est indiqué par l'identificateur d'objet du type d'attribut.

6.3.3.3 Classe d'objets informationnels ATTRIBUTE (attribut)

Un type d'attribut est défini comme une instance de la classe d'objets informationnels ATTRIBUTE. Cette classe est définie par un ensemble de spécifications de champs nommés, correspondant chacun à une des propriétés des instances de la classe. Ces champs sont les suivants:

- a) un identificateur **identifiant** pour l'attribut qui indique le type d'attribut;
- b) la syntaxe d'attribut **attribute-syntax** à laquelle se conformera chacune des valeurs de l'attribut;
- c) un ensemble de règles de criblage **matching-rules** qui identifient les types de cribles pouvant être appliqués à l'attribut (voir 6.3.9);
- d) une indication permettant de savoir si un attribut de ce type ne doit contenir qu'une seule valeur ou s'il peut en contenir plus d'une, toutes du même type ASN.1.

La syntaxe définie pour spécifier les instances de la classe d'objets informationnels ATTRIBUTE figure ci-dessous. Celle-ci spécifie une notation pour la définition des types d'attribut qui est utilisée tout au long de la présente Définition de service.

```

ATTRIBUTE ::= CLASS {
    &id
    &Type,
    &equalityMatch
    &substringsMatch
    &orderingMatch
    &numeration
        -- extension de 1994 --
    &OtherMatches
WITH SYNTAX {
    WITH ATTRIBUTE-SYNTAX
    [EQUALITY MATCHING-RULE
    [SUBSTRINGS MATCHING-RULE
    [ORDERING MATCHING-RULE
    [OTHER MATCHING-RULES
    NUMERATION
    ID
AttributeType UNIQUE,
MATCHING-RULE OPTIONAL,
MATCHING-RULE OPTIONAL,
MATCHING-RULE OPTIONAL,
ENUMERATED {single-valued(0), multi-valued(1)},
MATCHING-RULE OPTIONAL }
    &Type,
    &equalityMatch,]
    &substringsMatch,]
    &orderingMatch,]
    &OtherMatches,]
    &numeration,
    &id }

```

Les spécifications de champ ont la signification suivante:

- le champ de valeur **&id** est le champ d'identification de la classe. Il permet de distinguer un type d'attribut donné de tous les autres types de la classe;
- le champ de type **&Type** indique la syntaxe de l'attribut à laquelle se conformera chacune des valeurs du type d'attribut;
- le champ de valeur **&equalityMatch** identifie la règle de criblage servant à évaluer la composante de relation d'égalité d'un élément de filtre lorsqu'il est appliqué à l'attribut; voir 8.1.2.2;
- le champ de valeur **&substringsMatch** identifie la règle de criblage servant à évaluer la composante de comparaison de sous-chaînes d'un élément de filtre lorsqu'il est appliqué à l'attribut; voir 8.1.2.2;
- le champ de valeur **&orderingMatch** identifie la règle de criblage servant à évaluer les composantes de relations d'ordre (supérieur ou égal, inférieur ou égal) d'un élément de filtre lorsqu'il est appliqué à l'attribut; voir 8.1.2.2;
- le champ d'ensemble de valeurs **&OtherMatches** identifie toutes les autres règles de criblage définies pour le type d'attribut;
- le champ de valeur **&numeration** spécifie si le type d'attribut est mono-valué ou multi-valué.

6.3.4 Entrées principales, entrées mères et entrées filles

Bien qu'elles soient généralement indépendantes les unes des autres, certaines entrées sont liées. Une **entrée fille** peut être rattachée à une autre, son **entrée mère**, dans une relation arborescente. Une entrée qui n'est pas une **entrée fille** est dite **entrée principale**.

Cette relation est définie au moyen de deux attributs généraux particuliers:

- parent-sequence-number** (numéro de séquence d'entrée mère): cet attribut mono-valué donne le numéro de séquence de l'entrée mère d'une entrée fille. Il ne figure pas dans une entrée principale. Cette définition est donnée au 11.2.51;
- child-sequence-numbers** (numéros de séquence d'entrée fille): cet attribut multi-valué donne les numéros de séquence de toutes les entrées filles d'une entrée mère. Il ne figure pas dans une entrée qui n'est pas une entrée mère. Cette définition est donnée au 11.2.8.

Les opérations abstraites du service abstrait de mémoire de messages (voir l'article 8) n'agissent par défaut que sur les entrées principales. Certaines peuvent être orientées pour agir sur toutes les entrées tant principales que filles. En particulier, l'argument de l'opération abstraite Delete (suppression) (voir 8.2.4) ne peut sélectionner que des entrées principales, ce qui entraîne la suppression de l'entrée principale et de toutes ses entrées descendantes.

Le seul cas de création d'une entrée fille défini dans la présente Définition de service est celui d'un rapport remis dans lequel figure un contenu renvoyé. L'entrée delivered-report (rapport remis) est l'entrée principale, et l'entrée returned-content (contenu renvoyé) est son entrée fille. D'autres spécifications définissant des types de contenu de message peuvent fixer des règles supplémentaires pour l'enregistrement du contenu des messages qui imposent l'utilisation d'une entrée principale et d'une ou plusieurs entrées filles qui lui sont liées.

6.3.5 Attributs propres au contenu

La présente Définition de service ne définit que des types d'attributs généraux qui modélisent des propriétés de messages, de messages d'essai et de rapports dont la signification est conservée indépendamment du type de contenu. Cependant, pour obtenir des informations sur le contenu d'un message, la mémoire de messages doit en connaître la syntaxe et la sémantique, comme l'indique le type de contenu. Les **attributs propres au contenu** sont les types d'attributs définis dans la spécification appropriée et se rapportant à un type de contenu donné.

Une instance particulière de la mémoire de messages peut connaître zéro (mémoire de messages «générique»), un ou plusieurs types de contenu. Lorsqu'une mémoire de messages rencontre un message dont elle ne connaît pas le type de contenu, elle n'est pas en mesure de créer des attributs propres au contenu dans l'entrée du message.

A l'exception du cas unique décrit au 6.3.4, les règles qui régissent l'utilisation d'entrées filles pour enregistrer les composantes du contenu du message sont propres à chaque type de contenu. Si les entrées filles sont utilisées de cette façon, un attribut synopsis propre au contenu peut fournir des renseignements sur ces entrées filles (par exemple taille, numéro de séquence).

Lorsque les règles propres au contenu définies pour un type de contenu figurant dans une entrée de renvoi de contenu imposent la création d'entrées filles, celles-ci se verront affecter le type d'entrée des messages remis.

NOTES

1 Le type de contenu Interpersonal Messaging (messagerie de personne à personne) utilise des entrées filles pour enregistrer le contenu des messages (voir 19.2 de la Rec. UIT-T X.420 | ISO/CEI 10021-7). Un message de personne à personne qui contient une ou plusieurs parties de corps de message de personne à personne retransmis est représenté dans la mémoire de messages par une entrée principale et par une entrée fille pour chaque partie de corps du message de personne à personne. L'attribut général de **contenu** de l'entrée principale comprend le contenu dans son entier. Par conséquent, chaque partie de corps du message de personne à personne est conceptuellement présente à la fois dans sa propre entrée fille et dans l'entrée principale.

2 Une partie de corps de type IP-message (message de personne à personne) peut elle-même contenir d'autres messages de personne à personne encapsulés récursivement. La mémoire de messages modélise cette structure par l'allocation récursive des entrées filles.

3 La structure interne des contenus d'un message de personne à personne est définie par son attribut de synopsis de messagerie de personne à personne. Ce type d'attribut est un exemple d'attribut synopsis propre au contenu.

6.3.6 Types d'entrée

Chaque entrée appartient à l'un des **types d'entrée** définis ci-dessous.

```
EntryType ::= INTEGER {
    delivered-message      (0),
    delivered-report      (1),
    returned-content      (2),
    -- extensions de 1994 --
    submitted-message     (3),
    submitted-probe       (4),
    draft-message         (5),
    auto-action-event     (6) }
```

Une entrée de type **delivered-message** (message remis) est créée par l'exécution de l'opération abstraite Message-delivery (remise de message) de l'accès de remise, et contient les informations associées au message remis. Les entrées de type **delivered-report** (rapport remis) et **returned-content** (contenu renvoyé) sont créées par l'exécution de l'opération abstraite Report-delivery (remise de rapport) de l'accès de remise, et représentent respectivement un rapport remis et le contenu renvoyé conditionnellement présent dans un rapport remis.

Une entrée de type **submitted-message** (message déposé) est créée par l'exécution de l'opération abstraite MS-message-submission (dépôt de message MS) de l'accès de dépôt MS, ou par l'exécution d'une action automatique qui fait appel à l'opération abstraite Message-submission (dépôt de message) de l'accès de dépôt MTS, et contient les informations correspondant au message déposé. Une entrée **submitted-probe** (message d'essai déposé) est créée par l'exécution de l'opération abstraite MS-probe-submission (dépôt de message d'essai MS) de l'accès de dépôt MS et contient les informations correspondant au message d'essai déposé.

Une entrée **draft-message** (brouillon de message) est créée par l'exécution de l'opération abstraite MS-message-submission (dépôt de message MS) de l'accès de dépôt MS, et représente un message qui n'a pas été déposé auprès du système de transfert de messages, mais qui est stocké en vue d'une extraction ou d'un dépôt éventuels. La création d'un brouillon est décidée par l'utilisateur MS pour chaque message qu'il dépose.

Une entrée **auto-action-event** (événement d'action automatique) est créée par l'exécution d'une action automatique par la mémoire de messages, et enregistre le résultat de l'exécution de cette action (voir 6.5). Une ou plusieurs entrées peuvent être créées chaque fois qu'une action automatique est exécutée par la mémoire de messages, et fournissent ainsi un journal des actions automatiques exécutées par la mémoire de messages pour le compte de l'utilisateur MS. La définition d'une action automatique spécifie si son exécution peut être consignée par la création d'une entrée auto-action-event (événement d'action automatique).

Une entrée delivered-message (message remis), returned-content (contenu renvoyé), submitted-message (message déposé), ou draft-message (brouillon de message) peut comporter un contenu de n'importe quel type.

Les types d'attributs généraux présents dans chacun des types d'entrée mentionnés sont énumérés dans les Tableaux 2 et 3 et définis à l'article 11.

6.3.7 Organisation des classes d'entrée

La classe d'objets informationnels représentée par chaque classe d'entrées et les types d'entrée qu'elle contient sont définis ci-dessous. Le Tableau 1 récapitule les types d'entrée présents dans chaque classe. Les types d'attributs généraux pris en charge par chaque classe d'entrées sont énumérés dans les Tableaux 2 et 3 et définis à l'article 11.

Tableau 1 – Types d'entrée présents dans les classes d'entrée

Classes d'entrée	Types d'entrée						
	delivered-message (message remis)	delivered-report (rapport remis)	returned-content (contenu renvoyé)	submitted-message (message déposé)	submitted-probe (message d'essai déposé)	draft-message (brouillon de message)	auto-action-event (événement d'action automatique)
Delivery (remise)	x	x	x	–	–	–	–
Submission (dépôt)	–	–	–	x	x	–	–
Draft (brouillon)	–	–	–	–	–	x	–
Stored-message (message enregistré)	x	x	x	x	x	x	–
Delivery-log (journal de remise)	x	x	–	–	–	–	–
Submission-log (journal de dépôt)	–	–	–	x	x	–	–
Message-log (journal de messagerie)	x	x	–	x	x	–	–
Auto-action-log (journal d'actions automatiques)	–	–	–	–	–	–	x
x type d'entrée présent							
– type d'entrée absent							

6.3.7.1 Classe d'entrées Stored-message (message enregistré)

La classe d'entrées **Stored-message** (message enregistré) contient des entrées qui représentent des messages, rapports, messages d'essai et brouillons complets pour un ensemble sans restriction de types de contenu. On définit trois classes d'entrée subordonnées qui contiennent des sous-ensembles des entrées figurant dans la classe d'entrées Stored-message (message enregistré):

- a) la classe d'entrées **Delivery** (remise) contient les entrées du type delivered-message (message remis), delivered-report (rapport remis) et returned-content (contenu renvoyé), qui sont créées par l'exécution des opérations abstraites Message-delivery (remise de message) et Report-delivery (remise de rapport) de l'accès de remise. L'utilisateur MS ne possède aucune commande sélective sur la création des entrées de la classe Delivery (remise);
- b) la classe d'entrées **Submission** (dépôt) contient les entrées du type d'entrée submitted-message (message déposé) et submitted-probe (message d'essai déposé), qui sont créées par l'exécution des opérations abstraites MS-message-submission (dépôt de message MS) et MS-probe-submission (dépôt de message d'essai MS), ou par l'exécution d'actions automatiques faisant appel à l'opération abstraite Message-submission (dépôt de message). La mémoire de messages ne créera une entrée submitted-message

(message déposé) ou submitted-probe (message d'essai déposé) que si le dépôt est efficace, et si l'utilisateur MS le demande. La demande peut être faite explicitement dans les arguments de dépôt ou les paramètres de consignation de l'action automatique, ou peut être consignée pour tous les dépôts au moyen de l'opération abstraite Register-MS (consignation MS);

- c) la classe d'entrées **Draft** (brouillon) contient les entrées du type draft-message (brouillon de message). La valeur du paramètre submission-options (options de dépôt), facultativement présent dans l'argument de l'opération abstraite MS-message-submission (dépôt de message MS), commande le dépôt effectif du message (voir 8.1.6) ou la création d'une entrée de la classe brouillon (sans que celui-ci ne soit déposé auprès du système de transfert de messages). La principale différence entre les entrées de la classe Draft (brouillon) et celles de la classe Submission (dépôt) est que les premières ne possèdent pas les attributs dérivés du paramètre result (résultat) de l'opération abstraite Message-submission (dépôt de message) renvoyé par le système de transfert de messages (MTS) après un dépôt efficace.

6.3.7.2 Classe d'entrées Message-log (journal de messagerie)

La classe d'entrées **Message-log** (journal de messagerie) contient des entrées qui donnent une représentation restreinte des messages, des rapports et des messages d'essai pour des messages de n'importe quel type de contenu. Ces entrées, qui sont prévues pour durer longtemps, ne contiennent qu'un sous-ensemble des attributs présents dans les entrées correspondantes de la classe Stored-message (message enregistré); voir le Tableau 2.

On définit deux classes d'entrées subordonnées qui contiennent des sous-ensembles des entrées figurant dans la classe d'entrées Message-log (journal de messagerie):

- a) la classe **Delivery-log** (journal de remise) contient les entrées du type delivered-message (message remis) et delivered-report (rapport remis);
- b) la classe **Submission-log** (journal de dépôt) contient les entrées du type submitted-message (message déposé) et submitted-probe (message d'essai déposé).

Il existe un lien étroit entre les entrées des classes Message-log (journal de messagerie) et Stored-message (message enregistré), entre les entrées des classes Delivery-log (journal de remise) et Delivery (remise), et entre les entrées des classes Submission-log (journal de dépôt) et Submission (dépôt). Le lien entre ces classes appariées est établi par l'application des règles suivantes:

- a) à chaque entrée principale de la classe Stored-message (message enregistré) correspond une entrée de la classe Message-log (journal de messagerie) ayant des attributs sequence-number (numéro de séquence) et creation-time (heure de création) identiques. Cependant, une entrée de la classe Message-log peut être conservée après la suppression de l'entrée correspondante de la classe Stored-message. Une entrée de la classe Message-log est créée en même temps que l'entrée correspondante de la classe Stored-message;
- b) une entrée de la classe Submission-log (journal de dépôt) sera créée lors du dépôt d'un message ou d'un message d'essai, indépendamment du succès ou de l'échec du dépôt, et indépendamment de la création ou non d'une entrée de la classe Submission (dépôt); voir 6.3.7.1. Si le dépôt a échoué, l'entrée de la classe Submission-log ne contiendra pas les attributs qui dérivent du résultat des opérations abstraites Message-submission (dépôt du message) ou Probe-submission (dépôt de message d'essai), mais contiendra un attribut MS-submission-error (erreur de dépôt MS) pour indiquer le motif de l'échec. L'utilisateur MS ne possède aucune commande sélective agissant sur la création d'entrées des classes Submission-log et Delivery-log (journal de remise);
- c) chaque attribut d'une entrée de la classe Message-log (journal de messagerie) – à l'exception des attributs deletion-time (heure de suppression) et MS-submission-error (erreur de dépôt MS), qui sont propres aux entrées de la classe Message-log – est identique à son attribut homologue dans l'entrée correspondante de la classe Stored-message (message enregistré). Les attributs des entrées de la classe Message-log ne seront pas soumis à des modifications directes – par l'opération abstraite Modify (modification) ou par l'action automatique Auto-modify (modification automatique) – à l'exception de l'attribut message-group-name (nom du groupe de messages) qui peut être modifié s'il n'existe aucune entrée correspondante de la classe Stored-message. Cependant, lorsqu'un attribut d'une entrée de la classe Stored-message est modifié, la modification apparaît également dans l'entrée correspondante de la classe Message-log;
- d) certains attributs présents dans des entrées de la classe Stored-message (message enregistré) ne figureront pas dans les entrées correspondantes de la classe Message-log (journal de messagerie). Il s'agit de: message-delivery-envelope (enveloppe de remise de messages), message-submission-envelope (enveloppe de dépôt de messages), probe-submission-envelope (enveloppe de dépôt de messages d'essai), report-delivery-envelope (enveloppe de remise de rapports) et content (contenu). L'ensemble des types d'attributs disponibles dans les entrées de la classe Message-log n'est pas imposé; il est déterminé par abonnement;

- e) une entrée de la classe Message-log (journal de messagerie) ne pourra être supprimée qu'après la suppression de toute entrée correspondante de la classe Stored-message (message enregistré). Cependant, selon la politique locale, la suppression d'entrées de la classe Message-log peut être soumise à des restrictions supplémentaires. La mémoire de messages continuera à mettre à jour les attributs qui renferment les informations de corrélation d'une entrée de la classe Message-log même après que l'entrée correspondante de la classe Stored-message a été supprimée;
- f) l'attribut deletion-time (heure de suppression) d'une entrée de la classe Message-log (journal de messagerie) est créé lorsque l'entrée correspondante de la classe Stored-message (message enregistré) est supprimée. Pour une entrée de la classe Submission-log (journal de dépôt) pour laquelle aucune entrée de la classe Submission (dépôt) n'est créée, l'attribut deletion-time (heure de suppression) est créé lorsque l'entrée de la classe Submission-log est créée, et se voit affecter la même valeur que l'attribut creation-time (heure de création) de l'entrée;
- g) d'autres spécifications définissant les types de contenu des messages peuvent établir des règles concernant la création d'une entrée fille dans la classe Message-log (journal de messagerie) pour chaque entrée fille présente dans une entrée de la classe Stored-message (message enregistré). De telles règles ne sont pas définies dans la présente Définition de service. En particulier, la classe d'entrées Message-log ne contiendra pas d'entrée fille représentant le contenu renvoyé d'un rapport remis.

NOTE – Les entrées des classes Draft (brouillon) et Auto-action-log (journal d'actions automatiques) ne sont pas liées aux entrées de la classe Message-log (journal de messagerie).

6.3.7.3 Classe d'entrées Auto-action-log (journal d'actions automatiques)

La classe d'entrées **Auto-action-log** (journal d'actions automatiques) contient des entrées d'un type unique qui enregistrent le résultat de l'exécution des actions automatiques; voir 6.5. Aucune classe d'entrées subordonnée n'est définie. La définition d'un type d'action automatique spécifie si elle peut générer des entrées dans la classe Auto-action-log (journal d'actions automatiques). Selon les options d'abonnement, la mémoire de messages peut consigner l'exécution de toutes les actions automatiques ou uniquement de celles qui se traduisent par une erreur d'action automatique, ou ne rien consigner; voir 6.5.3.

6.3.7.4 Prise en charge des classes d'entrée

Lorsqu'on utilise un contexte d'application 1988, seule la classe d'entrées Delivery (remise) sera disponible.

Lorsqu'on utilise un contexte d'application 1994, la prise en charge de la classe d'entrées Delivery (remise) est obligatoire. La prise en charge d'une classe d'entrées subordonnée suppose la prise en charge de la classe d'entrées à laquelle elle est subordonnée. Par conséquent, la prise en charge de la classe d'entrées Stored-message (message enregistré) est également obligatoire (mais si aucune autre classe d'entrées n'est prise en charge, elle ne contiendra que les entrées également présentes dans la classe Delivery). De même, la classe Message-log (journal de messagerie) sera prise en charge si la classe Submission-log (journal de dépôt) ou Delivery-log (journal de remise) est prise en charge. La classe Submission-log sera prise en charge si les classes Submission (dépôt) et Delivery-log (journal de remise) sont toutes deux prises en charge. La classe Delivery-log sera prise en charge si la classe Submission-log est prise en charge.

La disponibilité des classes d'entrées optionnelles dépend de l'abonnement.

NOTE – La prise en charge d'une action automatique n'implique pas en elle-même la prise en charge de la classe d'entrées Auto-action-log (journal d'actions automatiques).

6.3.8 Etat d'extraction

Une propriété importante de chaque entrée dans n'importe quelle classe d'entrées est son état d'extraction **retrieval-status**.

```
RetrievalStatus ::= INTEGER {
    new          (0),
    listed       (1),
    processed    (2) }
```

Les valeurs de l'état d'extraction sont les suivantes:

- a) *new* (nouveau): l'entrée n'a été ni listée par l'utilisateur MS ni traitée automatiquement par la mémoire de messages;
- b) *listed* (listé): les informations concernant l'entrée ont été renvoyées à l'utilisateur MS par une opération abstraite List (listage) ou Fetch (recherche), mais l'entrée n'a pas encore été entièrement *traitée*;

- c) *processed* (traité): l'utilisateur MS a «entièrement recherché» l'entrée, ou la mémoire de messages a effectué sur cette entrée une action automatique dont la définition entraîne le positionnement de l'état d'extraction à *processed* (traité). (A noter que certaines actions automatiques conduisent à la suppression du message.) La signification de l'expression «entièrement recherché» dépend du type de contenu du message et est définie dans la Spécification appropriée.

L'état d'extraction d'une entrée *delivered-report* (rapport remis) devient *processed* (traité) lorsque le paramètre *report-delivery-envelope* (enveloppe de remise du rapport) est extrait. L'état d'extraction d'une entrée de la classe *Auto-action-log* (journal d'actions automatiques) devient *processed* lorsque tous ses attributs sont extraits.

L'état d'extraction d'une entrée fille est géré selon les mêmes règles qu'une entrée principale. La modification de l'état d'extraction d'une entrée peut être le résultat d'opérations effectuées sur l'entrée mère ou fille de cette entrée. Si une entrée fille est logiquement présente dans un attribut de son entrée mère, l'extraction de cet attribut est considérée comme équivalant à l'extraction de tous les attributs de cette entrée fille. Si un attribut d'entrée est présent logiquement dans une ou plusieurs entrées filles, l'extraction de toutes ces entrées équivaut à celle de cet attribut.

L'extraction d'attributs d'une entrée de la classe *Message-log* (journal de messagerie) n'affectera pas l'état d'extraction de cette entrée ou de l'entrée correspondante dans la classe *Stored-message* (message enregistré). L'état d'extraction d'une entrée de la classe *Message-log* ne sera pas affecté par la suppression de l'entrée correspondante dans la classe *Stored-message*.

NOTE – L'expression *entry-status* (état de l'entrée) utilisée dans la Rec. X.413 du CCITT (1988) et (1992) | ISO/CEI 10021-5:1990 a été remplacée par l'expression *retrieval-status* (état d'extraction) dans la présente Définition de service.

6.3.9 Règles de criblage

Une des capacités importantes du service abstrait de mémoire de messages est de pouvoir sélectionner un ensemble d'entrées appartenant à une classe donnée sur la base d'assertions relatives à la valeur des attributs de ces entrées.

Une règle de criblage permet de sélectionner des entrées en énonçant une assertion particulière concernant leurs valeurs d'attribut.

Le type d'assertion le plus simple est le type *attribute-value-assertion* (assertion sur la valeur de l'attribut) (voir 8.1.2.3). Des assertions plus complexes peuvent être prises en charge en utilisant des assertions de type *matching-rule-assertion* (assertion de règle de criblage). Une assertion de règle de criblage est une proposition qui peut être *vraie*, *fausse* ou *indéfinie*, relative à la présence dans une entrée d'une valeur d'attribut répondant aux critères définis par la règle de criblage.

Une assertion sur la valeur d'attribut ou une assertion de règle de criblage est évaluée d'après la règle de criblage associée à l'assertion.

Une règle de criblage est définie par la spécification:

- des types de cribles pris en charge par la règle;
- de la syntaxe requise pour exprimer une assertion pour chaque type de crible.

Certaines règles de criblage sont définies dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 et dans la Rec. UIT-T X.520 | ISO/CEI 9594-6 et à l'article 12. D'autres règles de criblage nécessaires à la prise en charge des attributs propres au contenu peuvent être définies dans les Spécifications de ces types de contenu; d'autres encore peuvent être définies par les autorités administratives nationales et les organisations privées.

NOTE – Aucune restriction n'est imposée aux règles de criblage qui peuvent être définies pour un type de contenu ou qui peuvent être définies localement. Cependant, les règles définies localement peuvent ne pas être connues des mémoires de messages et des agents utilisateur (UA). Chaque fois que cela est possible, il sera préférable d'utiliser les règles de criblage définies dans la présente Spécification ou dans d'autres Spécifications plutôt que d'en créer de nouvelles.

Parfois, une règle de criblage correspondra à la prise en charge d'un seul type de crible. Par exemple, le service abstrait de mémoire de messages prend en charge une relation de présence constituant un crible de présence.

Parfois, une règle de criblage correspondra à la prise en charge de plusieurs types de cribles. Par exemple, le service abstrait de mémoire de messages prend en charge une règle générique d'ordonnement autorisant les types de cribles «supérieur ou égal» et «inférieur ou égal».

6.3.9.1 Règles de criblage génériques

Il existe un certain nombre de règles de criblage génériques communes aux valeurs de nombreux types d'attributs différents. Ces règles sont:

- présence;
- égalité;
- sous-chaînes;
- ordonnancement;
- approximation.

La syntaxe permettant d'énoncer des assertions sur certains types de cribles associés à ces règles génériques a été définie dans le service abstrait de mémoire de messages:

- une syntaxe **present** (présence) pour le crible de présence;
- une syntaxe **equality** (égalité) pour le crible d'égalité;
- des syntaxes **initial** (initial), **any** (n'importe) et **final** (final) pour le crible de sous-chaînes;
- des syntaxes **greaterOrEqual** (supérieur ou égal) et **lessOrEqual** (inférieur ou égal) pour le crible d'ordonnancement;
- une syntaxe **approximateMatch** (criblage d'approximatif) pour le crible d'approximation.

La mémoire de messages prend en charge une règle de criblage de présence qui peut être appliquée à un attribut de n'importe quel type. Le crible de présence vérifie la présence de n'importe quelle valeur d'un type donné.

La mémoire de messages prend en charge une règle de criblage d'approximation qu'elle définit localement.

Quand un type d'attribut est défini, on indiquera les règles de criblage associées aux règles génériques d'égalité, de sous-chaînes, et d'ordonnancement prises en charge par le service abstrait de mémoire de messages. Ces règles servent lors de l'évaluation des assertions sur les relations d'égalité, d'ordonnancement et de sous-chaînes énoncées dans la syntaxe fournie par le service abstrait de mémoire de messages. Si aucune règle n'est définie, la seule assertion qui peut être énoncée concernant l'attribut porte sur sa présence.

Quand un criblage générique est effectué, la syntaxe de la valeur présentée sera identique à la syntaxe de l'attribut. Cependant, une règle de criblage peut être définie avec une syntaxe d'assertion qui est un choix (CHOICE) de types, de sorte qu'elle peut s'appliquer à des types d'attribut de syntaxes différentes (par exemple, différents types de chaînes de caractères). Par conséquent, les règles de criblage possibles qui peuvent être spécifiées comme règles génériques de criblage dans la définition d'un attribut se limitent à celles pour lesquelles la syntaxe d'attribut est un sous-type de la syntaxe d'assertion.

Une règle de criblage peut être utilisée à la fois comme une règle générique de criblage pour un type d'attribut et comme autre règle de criblage pour d'autres types d'attribut; voir 8.1.2.2.

6.3.9.2 Définition des règles de criblage

La définition d'une règle de criblage consiste à:

- a) affecter un identificateur d'objet à la règle de criblage;
- b) spécifier les différents types de cribles pris en charge par la règle;
- c) définir la syntaxe d'une assertion de la règle de criblage;
- d) définir les règles appropriées permettant d'évaluer une assertion en fonction des valeurs d'attribut cibles d'une entrée.

Une règle de criblage peut s'appliquer à différents types d'attributs possédant différentes syntaxes d'attribut.

La définition d'une règle de criblage comportera la spécification de la syntaxe des assertions de criblage, et la façon d'utiliser les valeurs de cette syntaxe pour procéder au criblage. Cette définition ne nécessite pas la spécification complète de la syntaxe d'attribut à laquelle la règle de criblage peut s'appliquer. La définition d'une règle de criblage applicable à des attributs obéissant à des syntaxes ASN.1 différentes spécifiera la façon d'effectuer le criblage. La syntaxe utilisée dans l'assertion d'une règle de criblage (c'est-à-dire pour la composante valeur de criblage de l'assertion) est la syntaxe d'assertion de la règle de criblage.

6.3.9.3 Classe d'objets informationnels MATCHING-RULE (règle de criblage)

Chaque règle de criblage définie dans la présente Définition de service est une instance de la classe d'objets informationnels MATCHING-RULE (règle de criblage), qui est définie dans la Rec. UIT-T X.501 | ISO/CEI 9594-2. Elle est reproduite ci-dessous pour faciliter la présentation:

```
MATCHING-RULE ::= CLASS {
    &AssertionType    OPTIONAL,
    &id                OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
    [SYNTAX    &AssertionType]
    ID        &id }
```

La classe d'objets contient les champs suivants:

- a) le champ de type **&AssertionType** indique la syntaxe d'assertion, c'est-à-dire la syntaxe à laquelle chaque valeur figurant dans une assertion de règle de criblage énoncée sera conforme. Ce champ est toujours présent dans les règles de criblage utilisées dans la présente Définition de service;
- b) le champ de valeur **&id** est le champ d'identification de la classe et contient un identificateur d'objet.

NOTES

- 1 Aucun support n'est fourni dans la syntaxe définie pour définir effectivement les règles de criblage elles-mêmes; cela doit être réalisé en langage naturel ou par un autre moyen.
- 2 La syntaxe définie sert à définir les règles de criblage de la présente Spécification et des autres Spécifications.

6.4 Groupage des messages

L'objet du **groupage des messages** est de fournir un cadre organisationnel qui permette à l'utilisateur d'établir un schéma de classification pour l'enregistrement des messages, accompagnant les moyens manuels ou automatiques d'affectation des messages aux catégories définies par l'utilisateur. Ceci permet à l'utilisateur de faire en sorte que les messages et messages d'essai déposés, les messages et rapports remis, et les brouillons de messages soient stockés de façon à pouvoir les localiser ultérieurement, par une sélection basée sur la classification décidée par l'utilisateur et adaptée à la nature de ses communications.

Le groupage des messages utilise le type d'attribut **message-group-name** (nom du groupe de messages) (voir 11.2.33) dont la syntaxe d'attribut est la suivante:

```
MessageGroupName ::= SEQUENCE SIZE(1..ub-group-depth) OF GroupNamePart
```

```
GroupNamePart ::= GeneralString (SIZE(1..ub-group-part-length))
```

Toute entrée dans la mémoire de messages, sauf les entrées de la classe Auto-action-log (journal d'actions automatiques), peut se voir attacher zéro, une ou plusieurs valeurs d'attribut message-group-name. Toutes les entrées possédant une même valeur d'attribut message-group-name peuvent être considérées comme appartenant à un même groupe de messages. La mémoire de messages offre les fonctionnalités suivantes pour prendre en charge le groupage des messages:

- a) l'utilisateur MS consignera dans la mémoire de messages, par l'opération abstraite Register-MS (consignation MS), l'ensemble des noms de groupe de messages qu'il envisage d'utiliser pour classer les messages. Cette opération est analogue à la création d'une structure de répertoires de fichiers. Ultérieurement, la mémoire de messages vérifiera que seuls les noms de groupe de messages consignés sont affectés aux entrées, pour garantir l'homogénéité interne. Un descripteur de groupe de messages peut accompagner chaque nom de groupe consigné pour en donner la description informelle du contenu; voir 8.2.5;
- b) la mémoire de messages ne permet la consignation d'un nom de groupe de messages comportant plus d'une partie de nom de groupe que si le nom de son ascendant (c'est-à-dire le nom de groupe de messages obtenu en en omettant la dernière partie) a antérieurement été consigné;
- c) la mémoire de messages permet la suppression d'un nom de groupe de messages consigné à condition qu'il ne figure dans l'attribut message-group-name (nom de groupe de messages) d'aucune entrée, qu'il ne soit l'ascendant d'aucun autre nom de groupe de messages consigné, et qu'il n'apparaisse pas dans la composante de modifications d'une action Auto-modify (modification automatique) consignée;

- d) la mémoire de messages peut être chargée d'attribuer automatiquement des noms de groupe appropriés aux messages, conformément à des critères de sélection spécifiés. Cette attribution est exécutée par l'action de modification automatique Auto-modify; voir 13.2;
- e) l'utilisateur MS peut modifier manuellement l'appartenance aux groupes de messages. Cette modification est exécutée par l'opération abstraite Modify (modification); voir 8.2.7;
- f) l'utilisateur MS peut extraire des détails de l'ensemble des noms de groupe de messages consignés et les afficher. Cette extraction est exécutée par l'opération abstraite Register-MS (consignation MS);
- g) l'appartenance à un groupe de messages peut être déterminée par une sélection portant sur la valeur de l'attribut message-group-name (nom de groupe de messages). Il n'est pas nécessaire que la mémoire de messages prenne des dispositions spéciales à cette fin, car la sélection des entrées sur la base de la valeur de leurs attributs est une fonction de base de la mémoire de messages.

NOTE – Cependant, des gains d'exécution peuvent découler d'une organisation de l'enregistrement des entrées reflétant l'appartenance aux groupes de messages.

L'Annexe H contient une description plus détaillée du groupage de messages et de ses utilisations.

6.5 Actions automatiques

Une **action automatique** est une action exécutée automatiquement par la mémoire de messages chaque fois que les conditions définies pour son exécution sont satisfaites. Ces conditions sont consignées par la mémoire de messages par abonnement, ou au moyen de l'opération abstraite Register-MS (consignation MS) (voir 8.2.5). Le résultat de l'exécution d'une action automatique est visible extérieurement pour la mémoire de messages.

Chaque type d'action automatique est identifié par son attribut **auto-action-type**. La définition de certaines actions automatiques spécifie un paramètre que la mémoire de messages utilisera en exécutant l'action automatique. Ce paramètre, **registration-parameter** (paramètre de consignation), est présenté à la mémoire de messages lorsque l'action automatique est consignée. Etant donné que certains types d'actions automatiques autorisent plus d'une consignation, chaque consignation est identifiée par un identificateur **registration-identifiant**. Les actions automatiques consignées d'un type donné sont exécutées par ordre croissant des identificateurs de consignation.

Quand un paramètre de consignation est défini pour un type d'action automatique, les informations qu'il contient peuvent être acheminées à la mémoire de messages par abonnement. Certaines actions automatiques peuvent cependant exiger que la mémoire de messages gère également la consignation de son paramètre de consignation au moyen de l'opération abstraite Register-MS (consignation).

Les **actions automatiques générales** s'appliquent aux entrées de tous types de contenu et sont définies à l'article 13. D'autres types d'actions automatiques propres à un type de contenu donné sont définis dans la Spécification de ce type de contenu.

6.5.1 Classe d'objets informationnels AUTO-ACTION (actions automatiques)

Chaque type d'action automatique est défini comme une instance de la classe d'objets informationnels AUTO-ACTION (actions automatiques):

```

AUTO-ACTION ::= CLASS {
    &id                               AutoActionType UNIQUE,
    &RegistrationParameter           OPTIONAL,
    &Errors                           AUTO-ACTION-ERROR OPTIONAL }
WITH SYNTAX {
    [REGISTRATION PARAMETER IS   &RegistrationParameter]
    [ERRORS                       &Errors]
    IDENTIFIED BY                 &id }

AutoActionType ::= OBJECT IDENTIFIER

```

La classe d'objets contient les champs suivants:

- a) le champ **&id** est le champ d'identification de la classe et contient un identificateur d'objet;
- b) le champ de type **&RegistrationParameter** définit la syntaxe du paramètre de consignation registration-parameter associé au type d'action automatique; voir 6.5.2. Ce champ est facultatif étant donné que certains types d'actions automatiques ne nécessitent pas de paramètre de consignation;
- c) le champ d'ensemble de valeurs **&Errors** définit un ensemble d'erreurs, chacune d'elles correspondant à l'échec d'exécution d'une action automatique; voir 6.5.3.

6.5.2 Consignation d'action automatique

Une action automatique est consignée par la mémoire de messages au moyen du paramètre **auto-action-registraton** (consignation d'action automatique) de l'opération abstraite Register-MS (consignation MS); voir 8.2.5.1 a).

```

AutoActionRegistration ::= SEQUENCE {
    auto-action-type          AUTO-ACTION.&id ({AutoActionTable}),
    registration-identifiser  [0] INTEGER (1..ub-per-auto-action) DEFAULT 1,
    registration-parameter   [1] AUTO-ACTION.&RegistrationParameter ({AutoActionTable}
                                {@auto-action-type}) OPTIONAL }

```

Les composantes du paramètre **auto-action-registraton** (consignation d'action automatique) sont les suivantes:

- auto-action-type** (type d'action automatique) (obligatoire): identificateur d'objet qui identifie le type d'action automatique. L'ensemble des contraintes Auto-action-table (tableau des actions automatiques) est défini à l'article 13;
- registration-identifiser** (identificateur de consignation) (optionnel): utilisé pour distinguer cette consignation d'action automatique des autres consignations possédant le même type d'action automatique. Cette composante régit également l'ordre d'exécution des actions automatiques consignées de ce type d'actions automatiques;
- registration-parameter** (paramètre de consignation) (optionnel): paramètre que la mémoire de messages doit utiliser lorsqu'elle exécute cette action automatique consignée.

6.5.3 Erreurs d'action automatique

La classe d'entrées Auto-action-log (journal d'actions automatiques) fournit un enregistrement des actions automatiques exécutées par la mémoire de messages (voir 6.3.7.3), et l'utilisateur MS peut en disposer s'il y est abonné. La définition d'un type d'action automatique spécifie si l'exécution de l'action peut se traduire par la création d'une entrée dans la classe Auto-action-log (journal d'actions automatiques). Lorsque l'exécution d'une action automatique se traduit par une erreur, un attribut auto-action-error (erreur d'action automatique) peut être attaché à l'entrée de la classe Auto-action-log associée à cette exécution (si l'exécution aboutit, l'attribut est absent). Dans la première association abstraite établie après l'apparition d'une erreur d'action automatique, la mémoire de messages informe l'utilisateur MS de l'erreur à l'aide du paramètre auto-action-error-indication (indication d'erreur d'action automatique) du résultat de rattachement MS. Ce paramètre indique le numéro de séquence de la première entrée de la classe Auto-action-log (journal d'actions automatiques), créée depuis l'établissement de la dernière association abstraite, dans laquelle figure un attribut auto-action-error (erreur d'action automatique).

Chaque type d'erreur d'action automatique est une instance de la classe d'objets informationnels AUTO-ACTION-ERROR (erreur d'action automatique) qui est identique à la classe d'objets informationnels ABSTRACT-ERROR (erreur abstraite):

```

AUTO-ACTION-ERROR ::= ABSTRACT-ERROR

```

Tandis que les erreurs abstraites sont définies dans le contexte du service des opérations distantes (ROS), et qu'elles sont signalées au-delà des frontières entre systèmes ouverts, les erreurs d'action automatique entraînent la création d'attributs auto-action-error (erreur d'action automatique) (voir 11.2.4) dans les entrées de la classe Auto-action-log (journal d'actions automatiques). Ce type d'attribut a la syntaxe suivante:

```

AutoActionError ::= SET {
    error-code              [0] AUTO-ACTION-ERROR.&errorCode ({AutoActionErrorTable}),
    error-parameter        [1] AUTO-ACTION-ERROR.&ParameterType ({AutoActionErrorTable}
                                {@error-code}) OPTIONAL }

```

L'ensemble des contraintes Auto-action-error-table (tableau des erreurs d'action automatique) est défini à l'article 13. La présente Définition de service ne définit pas les erreurs d'action automatique lorsque des erreurs abstraites identiques ont déjà été définies. Toute erreur abstraite peut être utilisée comme une erreur d'action automatique.

6.5.4 Exécution d'actions automatiques

Les actions automatiques sont exécutées de façon autonome par la mémoire de messages (éventuellement en l'absence d'une association abstraite avec l'utilisateur MS) quand les conditions données sont remplies. La première étape de ce processus est l'identification d'une entrée qu'un événement quelconque a mise en position d'être soumise à une action automatique. L'événement le plus courant qui déclenche l'exécution d'une action automatique est la création d'une entrée, suite par exemple à une opération abstraite Message-delivery (remise de message) ou Message-submission (dépôt de message). D'autres événements, comme l'expiration d'une temporisation, peuvent également déclencher l'exécution d'une action automatique.

La seconde étape intervenant dans l'exécution d'une action automatique est l'évaluation de critères de sélection présents optionnellement dans le paramètre de consignation registration-parameter de l'action automatique, et auxquels l'entrée candidate doit satisfaire avant que ne lui soit appliquée l'action automatique. Certains types d'actions automatiques autorisent la consignation par la mémoire de messages de plus d'un ensemble de paramètres de consignation registration-parameter pour l'action automatique. La définition de ces types d'actions automatiques détermine si la première action automatique exécutée pour laquelle l'entrée a satisfait les critères de sélection est la seule exécutée pour ce type d'action automatique, ou si toutes les actions automatiques consignées de ce type sont exécutées. L'ordre d'exécution des actions automatiques consignées d'un type donné est fixé par l'ordre croissant des valeurs de leurs composantes registration-identifier (identificateur de consignation). L'ordre d'exécution de chaque type d'action automatique défini est décrit aux articles 15 et 16.

6.6 Extensions MS

La présente Définition de service définit le fonctionnement de la mémoire de messages indépendamment des types de contenu des messages en présence. Pour accueillir les fonctions supplémentaires qui pourront s'avérer nécessaires à la prise en charge de types de contenu particuliers et à d'autres extensions, on définit la classe d'objets informationnels suivante:

MS-EXTENSION ::= TYPE-IDENTIFIER

Cette classe d'objets informationnels possède deux champs: un champ d'identification (un identificateur d'objet), et un champ de type qui définit le type ASN.1 de chaque instance de cette classe d'informations; voir l'Annexe A de la Rec. UIT-T X.681 | ISO/CEI 8824-2. Cette classe d'objets est utilisée lorsque la syntaxe abstraite de la mémoire de messages nécessite un endroit de stockage propre à chaque type de contenu possible; se reporter à la Spécification d'un type de contenu donné pour la définition de chaque instance d'utilisation de cette classe d'objets.

NOTE 1 – Par exemple, le paragraphe 19.5.2.2 de la Rec. UIT-T X.420 | ISO/CEI 10021-7 définit des options de dépôt propres au système de messagerie de personne à personne qui occupent les extensions de la mémoire de messages définies dans le paramètre MS-submission-options (options de dépôt MS); voir 8.1.6.

Les types suivants sont dérivés de la classe d'objets informationnels MS-EXTENSION (extension MS):

MSExtensionItem ::= INSTANCE OF MS-EXTENSION

MSExtensions ::= SEQUENCE SIZE(1..ub-extensions) OF MSExtensionItem

Lorsque l'argument d'une opération abstraite autre que MS-bind (rattachement MS) contient une composante MS-extension (extension MS) inconnue de la mémoire de messages, celle-ci générera une erreur d'extension MS; voir 9.12. Une extension MS propre à un type de contenu particulier sera ignorée lors du traitement d'une entrée d'un autre type de contenu. Lorsque le résultat d'une opération abstraite contient une composante MS-extension (extension MS) inconnue de l'utilisateur MS, celui-ci l'ignorera.

NOTE 2 – L'opération abstraite MS-bind (rattachement MS) signale les extensions qui ne sont pas prises en charge par son résultat, afin de permettre l'établissement de l'association abstraite même en présence d'une extension non prise en charge.

7 Opérations de rattachement et de détachement MS

7.1 Opération abstraite MS-bind (rattachement MS)

L'opération abstraite **MS-bind** (rattachement MS) rattache les accès MS de dépôt, d'extraction et d'administration de l'utilisateur MS (client) à la mémoire de messages (fournisseur). Le demandeur (du rattachement MS) est l'utilisateur MS, le répondeur étant la mémoire de messages elle-même. Les informations échangées dans l'argument et le résultat du rattachement MS resteront valides pour la durée de l'association abstraite. L'opération abstraite MS-bind (rattachement MS) est définie comme suit:

ms-bind ABSTRACT-OPERATION ::= {
 ARGUMENT **MSBindArgument**
 RESULT **MSBindResult**
 ERRORS **{ms-bind-error} }**

NOTE – Le service abstrait de mémoire de messages ne fournit pas de mécanisme explicite pour prendre en charge les associations multiples; l'effet de l'invocation simultanée d'opérations abstraites sur différentes associations abstraites n'est pas défini. Lorsque les associations multiples sont prises en charge (sur choix local), il appartient à l'utilisateur MS de coordonner de telles invocations afin d'éviter les interférences mutuelles [par exemple, l'invocation simultanée d'opérations abstraites Modify (modification) pourrait échouer ou produire des résultats incontrôlés].

7.1.1 Argument de rattachement MS

Les paramètres de l'argument de rattachement MS **MS-bind-argument** sont utilisés pour identifier, authentifier et fixer le contexte de sécurité pour un utilisateur MS. Ils contiennent également un ensemble de restrictions applicables aux entrées à renvoyer comme résultat de l'opération abstraite Fetch (recherche), ils identifient un ensemble de consignations associées à cette instance de l'utilisateur MS, et ils peuvent contenir une demande pour être informé des classes d'entrée, des types d'actions automatiques, des types d'attributs, des règles de criblage, de la prise en charge des groupes de messages et des types de contenus auxquels l'utilisateur MS s'est abonné.

```
MSBindArgument ::= SET {
  initiator-name           ORAddressAndOrDirectoryName,
  initiator-credentials    [2] InitiatorCredentials,
  security-context         [3] IMPLICIT SecurityContext OPTIONAL,
  fetch-restrictions       [4] Restrictions OPTIONAL -- aucune restriction par défaut --,
  ms-configuration-request [5] BOOLEAN DEFAULT FALSE,
                          -- extensions de 1994 --
  ua-registration-identifie [6] RegistrationIdentifier OPTIONAL,
  bind-extensions          [7] MSExtensions OPTIONAL }
```

Les paramètres de l'argument de rattachement MS **MS-bind-argument** ont la signification suivante:

- initiator-name** (nom du demandeur) (conditionnel): ce paramètre, qui contient le nom de l'utilisateur MS, est défini plus avant au 8.1.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- initiator-credentials** (pouvoirs du demandeur) (obligatoire): ce paramètre contient les *pouvoirs* de l'utilisateur MS. Il peut être utilisé par la mémoire de messages pour authentifier l'identité de l'utilisateur MS (voir la Rec. UIT-T X.509 | ISO/CEI 9594-8).

Si on a recours à une authentification simple, les pouvoirs du demandeur comportent un mot de passe simple, dont la définition est donnée au 8.5.11 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Si on a recours à une authentification forte, les pouvoirs du demandeur comportent un jeton de rattachement de demandeur et, à titre facultatif, un certificat du demandeur. Le jeton de rattachement de demandeur et le certificat du demandeur sont définis au 8.1.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Les pouvoirs du demandeur de l'utilisateur MS peuvent différer de ceux qui sont utilisés dans le rattachement MTS comme l'explique 8.1.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;

- security-context** (contexte de sécurité) (optionnel): ce paramètre identifie le **contexte de sécurité** dans le cadre duquel l'utilisateur MS se propose d'agir. Le contexte de sécurité est défini au 8.1.1.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

Le contexte de sécurité comporte une ou plusieurs étiquettes de sécurité définissant la sensibilité en termes de sécurité des interactions qui peuvent intervenir entre l'utilisateur MS et la mémoire de messages pendant la durée de l'association abstraite, conformément à la politique de sécurité en vigueur. Le contexte de sécurité sera un de ceux qui sont autorisés par les étiquettes de sécurité utilisateur consignées de l'utilisateur MS et par les étiquettes de sécurité associées à la mémoire de messages.

En l'absence de ce paramètre, aucun contexte de sécurité n'est établi entre l'utilisateur MS et la mémoire de messages et la sensibilité en termes de sécurité des interactions qui peuvent intervenir entre l'utilisateur MS et la mémoire de messages est laissée au choix de celui qui invoque une opération abstraite;

- fetch-restrictions** (restrictions de recherche) (optionnel): ce paramètre spécifie les restrictions portant sur les entrées à renvoyer comme résultat de l'opération abstraite Fetch (recherche). Les **restrictions de recherche** restent en vigueur pour la durée de l'association abstraite et s'appliquent sauf si l'argument de recherche contredit explicitement les restrictions.

En l'absence de ce paramètre, la mémoire de messages ne doit pas appliquer de restriction de recherche pendant l'association abstraite.

```
Restrictions ::= SET {
  allowed-content-types    [0] SET SIZE (1..ub-content-types) OF OBJECT IDENTIFIER OPTIONAL
                          -- aucune restriction par défaut --,
  allowed-EITs             [1] MS-EITs OPTIONAL -- aucune restriction par défaut --,
  maximum-attribute-length [2] INTEGER OPTIONAL -- aucune restriction par défaut -- }
```

Le paramètre contient les composantes suivantes:

- 1) **allowed-content-types** (types de contenu autorisés) (conditionnel): types de contenu que l'utilisateur MS est prêt à accepter comme résultat de l'opération abstraite Fetch (recherche). Tout message ayant un type de contenu autre que ceux qui sont spécifiés ne sera pas renvoyé et donnera lieu à une erreur de restriction de recherche.

En l'absence de cette composante, la mémoire de messages n'appliquera pas de restriction de recherche aux types de contenus des entrées;

- 2) **allowed-EITs** (types d'information codée autorisés) (conditionnel): types d'information codée que l'utilisateur MS est prêt à accepter comme résultat de l'opération abstraite Fetch (recherche). Si un message contient des types d'information codée autres que ceux qui sont spécifiés, un filtrage a lieu afin de ne pas renvoyer les parties des types d'information codée non autorisés avec le texte du message. Si le message ne contient que des types d'information codée non autorisés, une erreur de restriction de recherche sera signalée.

MS-EITs ::= SET SIZE (1..ub-encoded-information-types) OF MS-EIT

MS-EIT ::= OBJECT IDENTIFIER

En l'absence de cette composante, la mémoire de messages n'appliquera pas de restriction de recherche aux types d'information codée des entrées;

NOTE 1 – La composante allowed-EITs (types d'information codée autorisés) est présente pour des motifs historiques et son utilisation est déconseillée. Les parties de corps présentes dans le contenu d'une entrée peuvent être indiquées par l'attribut de synopsis propre au contenu, et une réalisation n'a besoin de rechercher que les parties de corps qu'elle accepte.

- 3) **maximum-attribute-length** (longueur maximale de l'attribut) (conditionnel): longueur maximale en octets de l'attribut que l'utilisateur MS est prêt à accepter comme résultat de l'opération abstraite Fetch (recherche). Lorsque l'utilisateur MS tente de rechercher une valeur d'attribut dont la longueur de codage dépasse cette longueur maximale, la mémoire de messages renverra une erreur de restriction de recherche.

NOTE 2 – La taille en octets du codage d'une valeur d'attribut peut varier lorsque les règles de codage de base de l'ASN.1 autorisent plusieurs codages (par exemple primitif et structuré).

NOTE 3 – Dans les contextes d'application 1988, la restriction de longueur maximale d'attribut ne s'applique qu'à l'attribut de contenu de l'entrée.

En l'absence de cette composante, la mémoire de messages n'appliquera pas de restriction de recherche à la longueur des attributs;

- e) **MS-configuration-request** (demande de configuration MS) (conditionnel): ce paramètre spécifie si la mémoire de messages est requise de renvoyer les informations qui identifient les types d'actions automatiques, les types d'attributs, les classes d'entrée, les types de contenu, les règles de criblage et toute capacité supplémentaire auxquels l'utilisateur MS s'est abonné, et qui sont donc à disposition pendant l'association abstraite.

En l'absence de ce paramètre, la mémoire de messages ne renverra pas d'information de configuration MS dans le résultat du rattachement MS;

NOTE 4 – En pratique, un utilisateur MS peut masquer ces informations plutôt que de les demander à chaque rattachement MS. La mémoire de messages pourra indiquer un changement dans les services offerts à l'aide du paramètre service-information (information de service) du résultat de rattachement MS.

- f) **UA-registration-identif** (identificateur de consignation d'agent utilisateur) (optionnel): ce paramètre affecte un identificateur à un ensemble d'informations de consignation. Si une invocation antérieure de l'opération abstraite Register-MS (consignation MS) a enregistré un agent utilisateur avec cette même valeur d'identificateur, la consignation de cet agent s'appliquera pour la durée de la présente association abstraite. Si le paramètre n'identifie pas de consignation antérieure d'agent utilisateur, la mémoire de messages le signalera dans le résultat de rattachement MS. Si ce paramètre est absent, les valeurs générales par défaut (valeurs par défaut des attributs de listage et des attributs de recherche, et valeurs par défaut de dépôt) s'appliqueront si elles ont été préalablement consignées par l'opération abstraite Register-MS (consignation MS); voir 8.2.5.1.

RegistrationIdentifier ::= PrintableString
(SIZE(1..ub-ua-registration-identif-length))

- g) **bind-extensions** (extensions du rattachement) (optionnel): ce paramètre spécifie les extensions demandées par l'utilisateur MS; si la mémoire de messages prend en charge les extensions de rattachement demandées, celles-ci s'appliqueront pour la durée de l'association abstraite. Toute extension de rattachement non prise en charge par la mémoire de messages sera signalée dans le résultat du rattachement MS. Toute extension de rattachement incorrectement spécifiée sera signalée dans une erreur de rattachement MS. Les extensions associées à un type de contenu de message particulier peuvent être définies dans la Spécification appropriée. En l'absence de ce paramètre, aucune extension n'est demandée.

7.1.2 Résultat du rattachement MS

MSBindResult ::= SET {	
responder-credentials	[2] ResponderCredentials,
available-auto-actions	[3] SET SIZE (1..ub-auto-actions) OF AUTO-ACTION.&id ({AutoActionTable}) OPTIONAL,
available-attribute-types	[4] SET SIZE (1..ub-attributes-supported) OF ATTRIBUTE.&id ({AttributeTable}) OPTIONAL,
alert-indication	[5] BOOLEAN DEFAULT FALSE,
content-types-supported	[6] SET SIZE (1..ub-content-types) OF OBJECT IDENTIFIER OPTIONAL, -- <i>extensions de 1994</i> --
entry-classes-supported	[7] SET SIZE (1..ub-entry-classes) OF EntryClass OPTIONAL,
matching-rules-supported	[8] SET SIZE (1..ub-matching-rules) OF OBJECT IDENTIFIER OPTIONAL,
additional-capabilities	[9] MSExtensions OPTIONAL,
message-group-depth	[10] INTEGER (1..ub-group-depth) OPTIONAL,
auto-action-error-indication	[11] AutoActionErrorIndication OPTIONAL,
unsupported-extensions	[12] SET SIZE (1..ub-extensions) OF OBJECT IDENTIFIER OPTIONAL,
ua-registration-id-unknown	[13] BOOLEAN DEFAULT FALSE,
service-information	[14] GeneralString (SIZE (1..ub-service-information-length)) OPTIONAL }

Les paramètres du résultat du rattachement MS **MS-bind-result** ont la signification suivante:

- a) **responder-credentials** (pouvoirs du répondeur) (obligatoire): ce paramètre contient les pouvoirs de la mémoire de messages répondante. L'utilisateur MS peut s'en servir pour authentifier l'identité de la mémoire de messages (voir la Rec. UIT-T X.509 | ISO/CEI 9594-8).

Si on a recours à une authentification simple, les pouvoirs du répondeur comportent un mot de passe simple associé à la mémoire de messages. Le mot de passe est défini au 8.5.11 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Si on a recours à une authentification forte, les pouvoirs du répondeur comportent un jeton de rattachement du répondeur créé par la mémoire de messages. Le jeton de rattachement du répondeur est défini au 8.1.1.1.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;

- b) **available-auto-actions** (actions automatiques disponibles) (conditionnel): ce paramètre identifie l'ensemble des types d'action automatique auxquels l'utilisateur MS s'est abonné. Ce paramètre sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai* et si au moins un type d'action automatique est disponible; sinon, il est absent. L'ensemble des contraintes Auto-action-table (tableau des actions automatiques) est défini à l'article 13;
- c) **available-attribute-types** (types d'attributs disponibles) (conditionnel): ce paramètre identifie l'ensemble des types d'attributs auxquels l'utilisateur MS s'est abonné. Ce paramètre sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai*; sinon, il est absent;

NOTE 1 – Les types d'attribut disponibles doivent tous être identifiés par ce paramètre, y compris ceux dont la prise en charge est obligatoire. L'ensemble des types d'attribut dont la prise en charge est obligatoire peut changer au cours du temps.

- d) **alert-indication** (indication d'avertissement) (conditionnel): s'il a la valeur *vrai*, ce paramètre indique qu'une situation d'avertissement s'est produite depuis la dernière **indication d'avertissement** aboutie;

- e) **content-types-supported** (types de contenus pris en charge) (conditionnel): ce paramètre identifie l'ensemble des types de contenu pour lesquels l'utilisateur MS s'est abonné et que la mémoire de messages prend en charge, comme cela est défini dans la Spécification appropriée des types de contenu. Un message dont le type de contenu est absent de cet ensemble peut quand même être déposé ou remis puis extrait. Dans ce cas, aucun des types d'attribut ou des types d'action automatique propres à ce type de contenu ne sera à la disposition de l'utilisateur MS.

Ce paramètre sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai* et si au moins un type de contenu est pris en charge. Sinon, il est absent;

- f) **entry-classes-supported** (classes d'entrée prises en charge) (conditionnel): ce paramètre identifie les classes d'entrée prises en charge par la mémoire de messages. La prise en charge d'une classe d'entrées suppose qu'elle soit à la disposition des opérations abstraites Summarize (récapitulation), List (listage), Fetch (recherche), Delete (suppression) et Modify (modification), et des types d'action automatique pris en charge par la mémoire de messages. De plus, la prise en charge de la classe d'entrées Submission (dépôt) suppose celle de son utilisation dans les opérations abstraites MS-message-submission (dépôt de message MS) et MS-probe-submission (dépôt de message d'essai MS). La prise en charge de la classe d'entrées Draft (brouillon) suppose celle de son utilisation dans l'opération abstraite MS-message-submission. Ce paramètre sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai*; sinon il est absent;
- g) **matching-rules-supported** (règles de criblage prises en charge) (conditionnel): ce paramètre identifie les règles de criblage prises en charge par la mémoire de messages. La prise en charge d'une règle de criblage suppose la possibilité de l'utiliser avec un attribut pour lequel la règle est définie, lorsque ce type d'attribut est également pris en charge. Ce paramètre sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai* et si une règle de criblage au moins est prise en charge; sinon il est absent;

NOTE 2 – Toutes les règles de criblage prises en charge seront identifiées par ce paramètre, y compris celles dont la prise en charge est obligatoire. L'ensemble des règles de criblage dont la prise en charge est obligatoire peut changer au cours du temps.

- h) **additional-capabilities** (capacités supplémentaires) (conditionnel): ce paramètre indique les capacités supplémentaires, éventuellement propres au contenu, prises en charge par la mémoire de messages. Il sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai* et si une capacité supplémentaire au moins est prise en charge; sinon il est absent;
- i) **message-group-depth** (profondeur du groupe de messages) (conditionnel): ce paramètre indique le nombre maximal de parties de nom de groupe qui peuvent figurer dans un nom de groupe de messages. Il sera présent si le paramètre MS-configuration-request (demande de configuration MS) de l'argument de rattachement MS est positionné à *vrai* et si la mémoire de messages prend en charge le type d'attribut message-group-name (nom de groupe de messages); sinon il est absent. Si l'utilisateur MS présente, dans l'argument d'une opération abstraite ou dans le paramètre de consignation registration-parameter d'une action automatique, un nom de groupe de messages comportant un nombre de parties plus grand que le nombre spécifié par ce paramètre, il en résulte une erreur de groupe de messages;
- j) **auto-action-error-indication** (indication d'erreur d'action automatique) (conditionnel): ce paramètre sera présent si une action automatique exécutée par la mémoire de messages depuis l'établissement de la précédente association abstraite a entraîné la génération d'une erreur d'action automatique. Comme cette information ne peut pas être signalée dans un contexte d'application 1988, l'établissement d'une telle association n'affectera pas le positionnement de ce paramètre, qui sera présenté lors de l'utilisation suivante d'un contexte d'application 1994.

```
AutoActionErrorIndication ::= CHOICE {
    indication-only          [0] NULL,
    auto-action-log-entry    [1] SequenceNumber }
```

Si l'abonnement de l'utilisateur comporte la classe d'entrées Auto-action-log (journal d'actions automatiques), le paramètre indique le numéro de séquence de la première entrée consignante une erreur d'action automatique survenue depuis l'établissement de l'association abstraite précédente (établie dans un contexte d'application 1994). Sinon il contient la valeur néant. Le paramètre sera absent si aucune erreur d'action automatique ne s'est produite depuis l'établissement de l'association abstraite précédente;

- k) **unsupported-extensions** (extensions non prises en charge) (conditionnel): si l'utilisateur MS a fourni un paramètre `bind-extensions` (extensions du rattachement) dans l'argument de rattachement MS et si une ou plusieurs de ces extensions ne sont pas prises en charge par la mémoire de messages, ce paramètre identifie toutes les extensions non prises en charge. Le paramètre contient les valeurs des identificateurs d'objet de chacune des extensions non prises en charge, déduites du paramètre `bind-extensions`. Il est présent si le paramètre `bind-extensions` de l'argument de rattachement MS est présent et si au moins une des extensions n'est pas prise en charge; sinon il est absent;
- l) **UA-registration-id-unknown** (identificateur inconnu de consignation d'agent utilisateur) (conditionnel): ce paramètre est présent et a la valeur *vrai* si l'argument de rattachement MS spécifie un identificateur de consignation d'agent utilisateur qui n'a pas été préalablement consigné par l'opération abstraite `Register-MS` (consignation MS). Sinon il est absent;
- m) **service-information** (information de service) (conditionnel): ce paramètre fournit des informations concernant l'exploitation du service. Une réalisation de mémoire de messages peut utiliser ce paramètre pour signaler des informations opérationnelles comme la disponibilité du service ou l'utilisation des ressources. La présence du paramètre dépendra du fournisseur du service MS.

7.1.3 Erreur de rattachement MS

Une erreur de rattachement MS **MS-bind-error** signale un problème dans une tentative d'établissement d'une association abstraite. Elle est définie de la façon suivante:

```
ms-bind-error ABSTRACT-ERROR ::= {
  PARAMETER CHOICE {
    unqualified-error  BindProblem,
                      -- extension de 1994 --
    qualified-error    SET {
      bind-problem      [0] BindProblem,
      supplementary-information [1] GeneralString (SIZE(1..ub-supplementary-info-length)) OPTIONAL,
      bind-extension-errors [2] SET SIZE(1..ub-extensions) OF OBJECT IDENTIFIER OPTIONAL } }

  BindProblem ::= ENUMERATED {
    authentication-error      (0),
    unacceptable-security-context (1),
    unable-to-establish-association (2),
    ... -- extension de 1994 --,
    bind-extension-problem    (3) }
```

Les paramètres de l'erreur de rattachement MS **MS-bind-error** sont les suivants:

- a) **unqualified-error** (erreur non qualifiée) (conditionnel): ce paramètre indique la nature de l'erreur. S'il indique une erreur d'authentification *authentication-error*, l'association abstraite ne peut pas être établie parce que les pouvoirs du demandeur ne sont pas acceptables ou sont mal spécifiés. S'il indique un contexte de sécurité inacceptable *unacceptable-security-context*, le répondeur ne peut pas accepter le contexte de sécurité proposé par le demandeur de l'association. S'il indique une incapacité à établir une association *unable-to-establish-association*, le répondeur a rejeté la tentative du demandeur visant à établir une association abstraite pour un motif non spécifié. S'il indique un problème lié à l'extension du rattachement *bind-extension-problem*, l'association abstraite ne peut pas être établie parce que le paramètre `bind-extensions` (extensions du rattachement) spécifié dans l'argument de rattachement MS contient une extension mal spécifiée;
- b) **qualified-error** (erreur qualifiée) (conditionnel): ce paramètre indique la nature de l'erreur, et peut fournir aussi des informations complémentaires **supplementary-information** qui explicitent l'erreur. Lorsqu'un ou plusieurs paramètres `bind-extensions` (extensions du rattachement) ont été mal spécifiés, les valeurs des identificateurs d'objet de ces extensions seront signalées dans les erreurs d'extension du rattachement **bind-extension-errors**.

7.2 Opération abstraite MS-unbind (détachement MS)

L'opération abstraite **MS-unbind** (détachement MS) met fin à l'association abstraite et a pour conséquence l'abandon de toutes les restrictions de recherche **fetch-restrictions** spécifiées dans l'argument de l'opération abstraite `MS-bind` (rattachement MS). Il n'y a ni argument, ni résultat, ni erreur associés à l'opération abstraite `MS-unbind`.

```
ms-unbind ABSTRACT-OPERATION ::= emptyUnbind
```

8 Opérations abstraites

Le présent paragraphe définit les capacités offertes par le service abstrait de mémoire de messages à un utilisateur aux accès d'extraction et de dépôt MS de la mémoire de messages. Ces capacités sont modélisées comme des opérations abstraites qui peuvent entraîner des erreurs abstraites lorsqu'elles sont invoquées. Les opérations abstraites disponibles sont définies au 8.2 pour l'accès d'extraction, et au 8.3 pour l'accès de dépôt. Les erreurs abstraites qu'elles peuvent entraîner font l'objet de l'article 9. La définition formelle des accès est donnée au 6.2.

8.1 Types de données communs utilisés dans les opérations abstraites

Ce paragraphe définit les types de données communs suivants qui interviennent dans plusieurs opérations abstraites définies à l'article 8:

- a) intervalle;
- b) filtre;
- c) sélecteur;
- d) sélection d'information d'entrée;
- e) information d'entrée;
- f) options de dépôt MS;
- g) résultats communs de dépôt.

8.1.1 Intervalle

Un paramètre **range** (intervalle) est utilisé pour sélectionner une séquence contiguë d'entrées d'une classe d'entrées spécifiée.

```
Range ::= CHOICE {
  sequence-number-range    [0] NumberRange,
  creation-time-range      [1] TimeRange }
```

```
NumberRange ::= SEQUENCE {
  from    [0] SequenceNumber OPTIONAL -- pas de borne inférieure si cette composante est omise --,
  to      [1] SequenceNumber OPTIONAL -- pas de borne supérieure si cette composante est omise -- }
```

```
TimeRange ::= SEQUENCE {
  from    [0] CreationTime OPTIONAL -- pas de borne inférieure si cette composante est omise --,
  to      [1] CreationTime OPTIONAL -- pas de borne supérieure si cette composante est omise -- }
```

```
CreationTime ::= UTCTime
```

Les composantes du paramètre **range** (intervalle) ont la signification suivante:

- a) **sequence-number-range** (intervalle des numéros de séquence) (conditionnel);
- b) **creation-time-range** (intervalle des heures de création) (conditionnel): ces deux paramètres identifient la séquence contiguë des entrées à sélectionner. La composante **sequence-number-range** est exprimée sous forme de numéros de séquence et la composante **creation-time-range** est exprimée sous forme d'heures de création. L'heure de création d'une entrée indique l'heure à laquelle la mémoire de messages a créé l'entrée. Les numéros de séquence des entrées successives sont toujours en ordre croissant, mais plusieurs entrées adjacentes peuvent avoir la même heure de création. Les composantes des paramètres **number-range** (intervalle de numéros) et **time-range** (intervalle horaire) ont la signification suivante:

- 1) **from** (à partir de) (optionnel): borne inférieure de l'intervalle.

En l'absence de cette composante, la valeur par défaut est «pas de borne inférieure»; la sélection commence à la première entrée (numéro de séquence le moins élevé) de la classe d'entrées spécifiée;

- 2) **to** (jusqu'à) (optionnel): borne supérieure de l'intervalle.

En l'absence de cette composante, la valeur par défaut est «pas de borne supérieure»; la sélection s'achève à la dernière entrée (numéro de séquence le plus élevé) de la classe d'entrées spécifiée.

Si la composante **from** est supérieure à la composante **to**, l'intervalle est alors défini dans l'ordre *décroissant*; sinon, l'intervalle est défini dans l'ordre *croissant* (voir 8.1.3). Dans un contexte d'application 1988, l'intervalle sera spécifié dans l'ordre croissant. Les bornes inférieure et supérieure sont incluses dans l'intervalle.

8.1.2 Filtres

8.1.2.1 Filtre

Un paramètre **filter** (filtre) consiste à appliquer un test à une entrée donnée, ce test étant ensuite vérifié ou non par l'entrée. Le paramètre **filter** est une expression ayant pour termes des assertions concernant la présence ou la valeur de certains attributs de l'entrée; il prend la valeur *Vrai* si et seulement si l'expression est vérifiée. Un paramètre **filter** peut être *vrai*, *faux* ou *indéfini*.

```
Filter ::= CHOICE {
    item      [0] FilterItem,
    and       [1] SET OF Filter,
    or        [2] SET OF Filter,
    not       [3] Filter }
```

Un filtre est soit un élément de filtre, soit une expression faisant intervenir des filtres plus simples liés par les opérateurs logiques **and** (et), **or** (ou) et **not** (non):

- un filtre qui est un élément **item** a la valeur de l'élément de filtre (c'est-à-dire *vrai*, *faux* ou *indéfini*);
- un filtre qui est l'intersection logique (**and**) d'un ensemble de filtres est *vrai* si cet ensemble est vide ou si chaque filtre composant est *vrai*; il est *faux* si au moins un filtre est *faux*; sinon, il est *indéfini* (c'est-à-dire lorsqu'un filtre au moins est *indéfini* et qu'aucun autre n'est *faux*);
- un filtre qui est la réunion logique (**or**) d'un ensemble de filtres est *faux* si cet ensemble est vide ou si chaque filtre est *faux*; il est *vrai* si au moins un filtre est *vrai*; sinon, il est *indéfini* (c'est-à-dire lorsqu'un filtre au moins est *indéfini* et qu'aucun filtre n'est *vrai*);
- un filtre qui est la négation logique (**not**) d'un filtre est *vrai* si ce filtre est *faux*; *faux* s'il est *vrai*; et *indéfini* si ce filtre est *indéfini*.

8.1.2.2 Élément de filtre

Un élément de filtre **filter-item** est une assertion sur la présence ou la valeur d'un attribut d'un type particulier de l'entrée soumise au test. Chaque assertion de ce type est *vraie*, *fausse* ou *indéfinie*.

```
FilterItem ::= CHOICE {
    equality          [0] AttributeValueAssertion,
    substrings       [1] SEQUENCE {
        type          ATTRIBUTE.&id ({AttributeTable}),
        strings       SEQUENCE OF CHOICE {
            initial    [0] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}),
            any        [1] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}),
            final      [2] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}) } },
    greater-or-equal [2] AttributeValueAssertion,
    less-or-equal    [3] AttributeValueAssertion,
    present          [4] ATTRIBUTE.&id ({AttributeTable}),
    approximate-match [5] AttributeValueAssertion,
    -- extension de 1994 --
    other-match      [6] MatchingRuleAssertion }
```

```
MatchingRuleAssertion ::= SEQUENCE {
    matching-rule    [0] MATCHING-RULE.&id ({MatchingRuleTable}),
    attribute-type   [1] ATTRIBUTE.&id,
    match-value      [2] MATCHING-RULE.&AssertionType ({MatchingRuleTable} {@matching-rule}) }
```

Chaque élément **filter-item** comporte un type d'attribut identifiant l'attribut testé.

Une assertion sur la valeur d'un attribut ne peut être évaluée que si elle a été correctement spécifiée. Si la mémoire de messages ne prend pas en charge le type d'attribut (ou si l'utilisateur MS ne s'y est pas abonné), ou si la mémoire de messages ne prend pas en charge la règle de criblage, ou si la valeur présentée n'est pas conforme à la syntaxe d'assertion de la règle de criblage, ou si la définition du type d'attribut n'inclut pas le type de crible demandé, alors l'opération abstraite ou l'action automatique qui contient l'assertion échouera; l'opération abstraite renverra une erreur d'attribut.

NOTE – La présentation d'une syntaxe d'assertion ou d'une syntaxe d'attribut non valide peut être détectée par le service de présentation et entraîner le rejet de l'unité de données protocolaire.

Les assertions sur les valeurs d'un attribut sont évaluées par les règles de criblage génériques (**equality**: égalité, **substrings**: sous-chaînes, **greater-or-equal**: supérieur ou égal, **less-or-equal**: inférieur ou égal, et **approximate-match**: criblage approximatif), ou par toute autre règle de criblage définie pour ce type d'attribut. Le crible **present** (présence) s'applique à tous les types d'attribut. Une règle de criblage utilisée sous forme générique doit avoir une syntaxe d'assertion compatible avec la syntaxe de la composante correspondante de l'élément **filter-item**. Une règle de criblage utilisée sous la forme «autre criblage» n'est pas soumise à une telle contrainte. Une règle de criblage peut être utilisée sous forme générique pour un type d'attribut et sous la forme «autre criblage» pour un autre type d'attribut.

Toutes les règles de criblage définies dans la présente Définition de service ont pour résultat *vrai* ou *faux*. Cependant, des règles de criblage définies ailleurs peuvent aussi avoir pour résultat *indéfini*.

Si l'élément **filter-item** est spécifié correctement, et s'il énonce une assertion de type:

- a) **equality** (égalité), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut à laquelle la règle de criblage s'applique et si la valeur présentée renvoie l'évaluation *vrai*;
- b) **substrings** (sous-chaînes), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut à laquelle la règle de criblage s'applique et si la valeur présentée dans le paramètre **strings** (chaînes) renvoie l'évaluation *vrai*. La signification des composantes du paramètre **strings** est donnée par la définition de la règle de criblage appropriée; voir au 12.2.3 l'exemple d'une règle de criblage par sous-chaîne;
- c) **greater-or-equal** (supérieur ou égal), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut à laquelle la règle de criblage **ordering** (ordonnancement) s'applique et si la valeur présentée renvoie l'évaluation *faux*, c'est-à-dire s'il existe une valeur de l'attribut qui est *supérieure ou égale* à la valeur présentée;
- d) **less-or-equal** (inférieur ou égal), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut à laquelle s'appliquent les règles de criblage **equality** (égalité) ou **ordering** (ordonnancement) et si la valeur présentée renvoie l'évaluation *vrai*, c'est-à-dire s'il existe une valeur de l'attribut qui est *inférieure ou égale* à la valeur présentée;
- e) **present** (présent), il prend la valeur *vrai* si, et seulement si, cet attribut est présent dans l'entrée;
- f) **approximate-match** (criblage approximatif), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut à laquelle la règle de criblage **approximate-match** (criblage approximatif) s'applique et si la valeur présentée renvoie l'évaluation *vrai*. La règle de criblage **approximate-match**, qui peut assurer les comparaisons phonétiques pour compenser les variations orthographiques, est définie localement: il n'y a aucune directive spécifique pour le criblage approximatif dans la présente Définition de service. Si la règle de criblage **approximate-match** n'est pas prise en charge, cet élément de filtre sera traité comme s'il spécifiait une assertion de type **equality** (égalité);
- g) **other-match** (autre criblage), il prend la valeur *vrai* si, et seulement si, il existe une valeur de l'attribut ayant le type d'attribut indiqué et pour laquelle la règle de criblage appliquée à cette valeur et à la valeur présentée renvoie l'évaluation *vrai*. Si la règle de criblage est l'une des règles spécifiées comme «autres règles de criblage» dans la définition du type d'attribut, et si elle est prise en charge par la mémoire de messages, alors le criblage procède conformément à la définition de la règle de criblage; sinon il peut en résulter une erreur d'attribut avec pour motif «criblage inapproprié». Au niveau local, une mémoire de messages peut prendre en charge des règles de criblage en plus de celles qui sont spécifiées dans la présente Définition de service. L'ensemble des contraintes MatchingRuleTable (tableau des règles de criblage) est défini à l'article 12.

8.1.2.3 Assertion de valeur d'attribut

Une assertion de valeur d'attribut **attribute-value-assertion** est une proposition, *vraie*, *fausse* ou *indéfinie*, concernant les valeurs d'une entrée. Elle est évaluée au moyen d'une règle de criblage spécifiée pour le type d'attribut et qui convient au contexte dans lequel est évaluée l'assertion. Elle fait intervenir un type d'attribut et une valeur d'attribut:

```
AttributeValueAssertion ::= SEQUENCE {
    attribute-type      ATTRIBUTE.&id ({AttributeTable}),
    attribute-value     ATTRIBUTE.&Type ({AttributeTable} {@attribute-type}) }
```

Si l'assertion de valeur d'attribut est spécifiée correctement (voir 8.1.2.2), son évaluation conduit à l'un des résultats suivants:

- a) *vrai*, si l'entrée contient un attribut du type indiqué, et dont l'une des valeurs correspond à la valeur d'attribut présentée;
- b) *indéfini*, si l'entrée contient un attribut du type indiqué, et que la règle de criblage spécifiée déclare le résultat *indéfini*;
- c) *faux*, sinon.

NOTE – Aucune des règles de criblage spécifiées dans la présente Définition de service ne renvoie l'évaluation *indéfini*. Cependant, des règles de criblage définies dans d'autres Spécifications pour des types de contenu donnés pourront le faire.

8.1.3 Sélecteur

Un paramètre **selector** (sélecteur) sert à sélectionner les entrées d'une classe donnée. La sélection se fait en trois étapes. Dans la première, l'ensemble de toutes les entrées de la classe peut être restreint à un ensemble contigu particulier en en spécifiant l'intervalle de variation. Dans la deuxième étape, les entrées de cet ensemble peuvent être sélectionnées en spécifiant un filtre auquel doivent satisfaire les entrées sélectionnées. Dans la troisième étape, une limite peut être fixée au nombre d'entrées ainsi sélectionnées. Si l'intervalle est défini dans l'ordre croissant (ou si ce paramètre est omis), les entrées portant les plus petits numéros de séquence seront sélectionnées; si l'intervalle est défini dans l'ordre décroissant, les entrées portant les plus forts numéros de séquence seront sélectionnées (voir 8.1.1). Les opérations abstraites sont appliquées aux entrées sélectionnées selon l'ordre croissant ou décroissant de définition de leur intervalle; c'est-à-dire que si l'intervalle est défini dans l'ordre croissant, l'entrée sélectionnée dont le numéro de séquence est le plus petit est traitée en premier.

```
Selector ::= SET {
  child-entries [0] BOOLEAN DEFAULT FALSE,
  range         [1] Range OPTIONAL -- pas de bornes par défaut --,
  filter        [2] Filter OPTIONAL -- par défaut, toutes les entrées de l'intervalle spécifié --,
  limit         [3] INTEGER (1..ub-messages) OPTIONAL,
  override      [4] OverrideRestrictions OPTIONAL
                -- par défaut, toute restriction de recherche en vigueur s'applique -- }
```

Les composantes du paramètre **selector** (sélecteur) ont la signification suivante:

- child-entries** (entrées filles) (optionnel): si la valeur de cette composante est *fausse*, seules les entrées principales sont prises en compte dans la sélection. Si elle est *vraie*, les entrées principales et les entrées filles sont prises en compte dans la sélection. En l'absence de cette composante, seules sont prises en compte les entrées principales;
- range** (intervalle) (optionnel): la notation de syntaxe abstraite de la composante range est donnée au 8.1.1. En l'absence de cette composante, l'intervalle n'est pas borné;
- filter** (filtre) (optionnel): la notation de syntaxe abstraite de la composante filter est donnée au 8.1.2.1. En l'absence de cette composante, toutes les entrées de l'intervalle spécifié sont sélectionnées;
- limit** (limite) (optionnel): cette composante permet de spécifier une limite supérieure au nombre d'entrées sélectionnées. En l'absence de cette composante, aucune limite n'est fixée à ce nombre;

NOTE – Le rôle essentiel de la limite est d'éviter qu'une opération abstraite avec une sélection mal formulée ne conduise à une avalanche de résultats. Cette limite peut également servir à extraire une quantité exacte d'informations convenant à tel ou tel dispositif de sortie.

- override** (ignorer) (optionnel): s'il faut ignorer une restriction de recherche **fetch-restrictions**, la ou les composantes correspondantes du paramètre **override-restrictions** (ignorer les restrictions) sera (seront) présentes.

```
OverrideRestrictions ::= BIT STRING {
  override-content-types-restriction (0),
  override-EITs-restriction          (1),
  override-attribute-length-restriction (2) } (SIZE (1..ub-restrictions))
```

Les valeurs du paramètre **override-restrictions** (ignorer les restrictions) ont la signification suivante:

- override-content-types-restriction** (ignorer la restriction sur les types de contenu) (obligatoire): ce bit sera positionné à 1 s'il faut ignorer les restrictions de recherche portant sur les types de contenus. Si ce bit est positionné à 0, les restrictions de recherche **fetch-restrictions** spécifiées dans l'argument de rattachement MS et portant sur les types de contenus s'appliqueront;
- override-EITs-restriction** (ignorer la restriction sur les types d'information codée) (obligatoire): ce bit sera positionné à 1 s'il faut ignorer les restrictions de recherche portant sur les types d'information codée. Si ce bit est positionné à 0, les restrictions de recherche **fetch-restrictions** spécifiées dans l'argument de rattachement MS et portant sur les types d'information codée s'appliqueront;
- override-attribute-length-restriction** (ignorer la restriction sur la longueur d'attribut) (obligatoire): ce bit sera positionné à 1 s'il faut ignorer les restrictions de recherche portant sur la longueur d'attribut. Si ce bit est positionné à 0, les restrictions de recherche **fetch-restrictions** spécifiées dans l'argument de rattachement MS et portant sur la longueur de l'attribut s'appliqueront.

En l'absence du paramètre **override-restrictions**, toutes les restrictions de recherche spécifiées dans l'argument de rattachement MS s'appliqueront.

8.1.4 Sélection d'information d'entrée

Un paramètre **entry-information-selection** (sélection d'information d'entrée) indique quelle information de l'entrée est demandée. Un ensemble vide indique qu'une information est demandée au sujet de l'entrée proprement dite, et non les attributs de celle-ci.

EntryInformationSelection ::= SET SIZE (0..ub-per-entry) OF AttributeSelection

AttributeSelection ::= SET {
type ATTRIBUTE.&id ({AttributeTable}),
from [0] INTEGER (1..ub-attribute-values) OPTIONAL -- utilisé si le type est multi-valué --,
count [1] INTEGER (1..ub-attribute-values) OPTIONAL -- utilisé si le type est multi-valué -- }

Les composantes du paramètre **attribute-selection** (sélection d'attribut) ont la signification suivante:

- type** (obligatoire): indique le type de l'attribut. Si aucun attribut de ce type n'est présent dans l'entrée, aucune valeur n'est renvoyée pour l'attribut. Si le type d'attribut n'est pas pris en charge par la mémoire de messages ou si l'utilisateur MS ne s'y est pas abonné, une erreur d'attribut sera générée;
- from** (à partir de) (optionnel): lorsqu'un attribut est multi-valué, cet entier donne la position relative de la première valeur à renvoyer. S'il spécifie une valeur supérieure à celles qui sont présentes dans l'attribut, aucune valeur n'est renvoyée. Cette composante ne sera présente que si l'attribut est multi-valué. Si elle est omise, les valeurs commençant à la première valeur sont renvoyées;
- count** (décompte) (optionnel): lorsqu'un attribut est multi-valué, cet entier donne le nombre maximal de valeurs à renvoyer. Le fait qu'il comporte une valeur supérieure au nombre de valeurs présentes dans l'entrée ne constitue pas une erreur. Cette composante ne sera présente que si l'attribut est multi-valué. Si elle est omise, le nombre de valeurs renvoyées n'est pas limité.

Si la composante **from** ou la composante **count** (ou les deux) sont spécifiées, la sélection d'attribut est une demande d'attribut partiel **partial-attribute-request**.

Si un attribut du type spécifié dans une sélection d'attribut est pris en charge par la mémoire de messages, mais qu'il ne figure pas dans une entrée donnée, ou qu'il ne s'applique pas à une entrée de ce type, on considérera la sélection comme absente lorsque l'opération abstraite associée est appliquée à l'entrée.

8.1.5 Information d'entrée

Un paramètre **entry-information** (information d'entrée) véhicule les informations sélectionnées provenant d'une entrée.

EntryInformation ::= SEQUENCE {
sequence-number SequenceNumber,
attributes SET SIZE (1..ub-per-entry) OF Attribute OPTIONAL,
 -- extension de 1994 --
value-count-exceeded [0] SET SIZE (1..ub-per-entry) OF AttributeValueCount OPTIONAL }

AttributeValueCount ::= SEQUENCE {
type [0] ATTRIBUTE.&id ({AttributeTable}),
total [1] INTEGER }

Les composantes du paramètre **entry-information** (information d'entrée) ont la signification suivante:

- sequence-number** (numéro de séquence) (obligatoire): numéro de séquence identifiant l'entrée (voir 6.3.2);
- attributes** (attributs) (conditionnel): ensemble des attributs sélectionnés de l'entrée. Lorsque la sélection d'attribut correspondante constitue une demande d'attribut partiel, seul le sous-ensemble demandé de valeurs effectivement présentes dans cet attribut est renvoyé. Cette composante est absente si le paramètre requested-attributes (attributs demandés) a été omis de l'argument de listage ou de recherche, et qu'aucune valeur par défaut d'attribut de recherche ou de listage (générale ou propre à l'agent utilisateur) n'est en vigueur. La composante est également absente si aucun des attributs demandés n'est présent dans l'entrée;

- c) **value-count-exceeded** (dépassement du décompte de valeurs) (conditionnel): cette composante est présente si, lors de l'exécution de l'opération abstraite List (listage) ou Fetch (recherche), la composante **count** (décompte) est spécifiée dans la sélection d'attribut, et si l'attribut spécifié contient un nombre de valeurs plus grand que le nombre fixé par la composante **count**. La composante **type** indique le type d'attribut concerné, et la composante **total** indique le nombre total de valeurs que l'attribut possède. La composante **value-count-exceeded** comportera une valeur pour chaque attribut sélectionné pour lequel la limite fixée par la composante **count** est dépassée. Cette composante sera absente si aucun attribut sélectionné n'excède la limite fixée par la composante **count**.

8.1.6 Options de dépôt MS

Le paramètre **MS-submission-options** (options de dépôt MS) est utilisé pour demander la création d'une entrée dans une classe d'entrées spécifiée, avec des attributs dérivés de l'argument et du résultat de l'opération abstraite associée MS-message-submission (dépôt de message MS) ou MS-probe-submission (dépôt de message d'essai MS). D'autres composantes de ce paramètre commandent l'affectation d'autres attributs à l'entrée créée.

```
MSSubmissionOptions ::= SET {
  object-entry-class           [0] EntryClass (submission | submission-log | draft) OPTIONAL,
  disable-auto-modify         [1] BOOLEAN DEFAULT FALSE,
  add-message-group-names     [2] SET SIZE (1..ub-message-groups) OF MessageGroupName OPTIONAL,
  ms-submission-extensions    [3] MSExtensions OPTIONAL }
```

Les composantes du paramètre **MS-submission-options** (options de dépôt MS) ont la signification suivante:

- a) **object-entry-class** (classe d'entrées de l'objet) (optionnel): cette composante spécifie la classe de l'entrée à créer, qui contiendra des détails sur le dépôt du message ou du message d'essai associé. Trois valeurs sont définies:
- 1) **submission** (dépôt). Le message ou le message d'essai sera déposé auprès du système MTS; une entrée sera créée dans la classe Submission, et, selon l'abonnement, une autre dans la classe Submission-log (journal de dépôt);
 - 2) **submission-log** (journal de dépôt). Le message ou le message d'essai sera déposé auprès du système MTS; une entrée sera créée dans la classe Submission-log (journal de dépôt) seulement;
 - 3) **draft** (brouillon). Une entrée sera créée dans la classe Draft (brouillon), mais le message ne sera pas déposé auprès du système MTS. Cette option n'existe pas pour le dépôt de message d'essai MS.

Si la composante **object-entry-class** (classe d'entrées de l'objet) spécifiée ne fait pas l'objet d'un abonnement, une erreur est déclarée. Si la composante **object-entry-class** est omise, et si la classe d'entrées Submission-log (journal de dépôt) fait l'objet d'un abonnement, la composante prend alors la valeur par défaut **submission-log**. Si la composante **object-entry-class** est omise, et si la classe Submission-log ne fait pas l'objet d'un abonnement, le message ou le message d'essai sera déposé auprès du système MTS et aucun enregistrement n'a lieu;

- b) **disable-Auto-modify** (mise hors service de la modification automatique) (optionnel): cette composante spécifie si les actions automatiques Auto-modify (modification automatique) actuellement consignées doivent être appliquées à l'entrée de la classe Submission (dépôt), Submission-log (journal de dépôt) ou Draft (brouillon), pouvant être créée à la suite de l'exécution du dépôt du message ou du message d'essai associé. Cette composante sera omise si aucune entrée ne doit être créée. En l'absence de cette composante, les actions de modification automatique consignées seront appliquées à la nouvelle entrée;
- c) **add-message-group-names** (noms de groupe de messages supplémentaires) (optionnel): cette composante spécifie une ou plusieurs valeurs de l'attribut message-group-name (nom de groupe de messages) qui seront affectées à l'entrée de la classe Submission (dépôt), Submission-log (journal de dépôt) ou Draft (brouillon), pouvant être créée à la suite de l'exécution du dépôt du message ou du message d'essai associé. Ce type d'attribut est défini au 11.2.33. Cette composante sera omise si aucune entrée ne doit être créée. En l'absence de cette composante, aucun attribut message-group-name n'est attaché à la nouvelle entrée;

NOTE – Les noms de groupe de messages affectés à une entrée par l'action automatique Auto-modify (modification automatique) ne sont pas affectés par le positionnement de cette composante.

- d) **ms-submission-extensions** (extensions du dépôt MS) (optionnel): cette composante permettra d'apporter ultérieurement aux options de dépôt MS des extensions générales ou propres au contenu. Aucune extension n'est définie dans la présente Définition de service. La Spécification pour un type de contenu donné définit l'utilisation de cette composante par la Spécification. En l'absence de cette composante, aucune extension n'est spécifiée.

8.1.7 Résultats communs de dépôt

Le paramètre **common-submission-results** (résultats communs de dépôt) contient des informations communes aux résultats des opérations abstraites MS-message-submission (dépôt de message MS) et MS-probe-submission (dépôt de message d'essai MS).

```
CommonSubmissionResults ::= SET {
  created-entry           [0] SequenceNumber OPTIONAL,
  auto-action-error-indication [1] AutoActionErrorIndication OPTIONAL,
  ms-submission-result-extensions [2] MSExtensions OPTIONAL }
```

Les composantes de ce paramètre ont la signification suivante:

- a) **created-entry** (entrée créée) (conditionnel): cette composante sera présente si l'utilisateur MS a demandé l'enregistrement d'un message ou d'un message d'essai déposé dans la classe d'entrées Submission (dépôt) ou Submission-log (journal de dépôt), ou l'enregistrement d'un brouillon de message dans la classe d'entrées Draft (brouillon). Elle indique le numéro de séquence de l'entrée nouvellement créée;
- b) **auto-action-error-indication** (indication d'erreur d'action automatique) (conditionnel): cette composante sera présente si une action automatique exécutée par la mémoire de messages concernant le message ou le message d'essai déposé, ou le brouillon de message enregistré, a entraîné la génération d'une erreur d'action automatique. Si la classe d'entrées Auto-action-log (journal d'actions automatiques) fait l'objet d'un abonnement, cette composante indique le numéro de séquence de la première entrée qui enregistre l'exécution des actions automatiques ayant échoué. Sinon elle contient la valeur néant; voir 7.1.2 j). Cette composante sera absente si aucune erreur d'action automatique ne se produit;
- c) **ms-submission-result-extensions** (extensions des résultats de dépôt MS) (conditionnel): cette composante permettra d'apporter ultérieurement aux résultats communs de dépôt des extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service. Elle sera présente si l'utilisateur MS a spécifié des extensions de dépôt MS dans le paramètre submission-options (options de dépôt) de l'argument de dépôt de message ou de dépôt de message d'essai MS, et si ces extensions ont entraîné la génération du résultat correspondant.

8.2 Opérations abstraites de l'accès d'extraction

Les opérations abstraites suivantes sont disponibles à l'accès d'extraction:

- a) Summarize (récapitulation);
- b) List (listage);
- c) Fetch (recherche);
- d) Delete (suppression);
- e) Register-MS (consignation MS);
- f) Alert (avertissement);
- g) Modify (modification).

L'exécution des opérations abstraites est asynchrone et soumise aux conditions suivantes:

- les opérations abstraites Delete (suppression), Register-MS (consignation MS), et Modify (modification) ne seront exécutées qu'une fois toutes les opérations abstraites en cours terminées;
- les opérations abstraites Delete (suppression), Register-MS (consignation MS), et Modify (modification) sont exécutées dans l'ordre dans lequel elles sont invoquées, et aucune autre opération abstraite ne peut être lancée avant qu'elles ne soient achevées.

Etant donné ces règles et le fait que les opérations abstraites List (listage) et Fetch (recherche) peuvent modifier l'état d'extraction d'une entrée, le résultat des opérations abstraites Summarize (récapitulation), List (listage) et Fetch (recherche) n'est pas nécessairement déterminé.

8.2.1 Opération abstraite Summarize (récapitulation)

L'opération abstraite **Summarize** (récapitulation) renvoie un récapitulatif des entrées sélectionnées d'une classe donnée. En plus de ces récapitulatifs, l'opération renvoie un décompte des entrées sélectionnées, ainsi que leurs numéros de séquence le plus petit et le plus grand. Zéro, un ou plusieurs récapitulatifs individuels peuvent être demandés.

L'opération abstraite **Summarize** n'aboutira que si le contexte de sécurité et la politique de sécurité en vigueur permettent l'accès à la classe d'entrées.

Les attributs qui peuvent être demandés dans les récapitulatifs sont limités. Ces restrictions sont définies dans les Tableaux 2 et 3 pour les attributs généraux des entrées des diverses classes. Quant aux attributs propres au contenu, leur définition dans la Spécification appropriée déterminera si l'attribut est récapitulable ou non.

```

summarize ABSTRACT-OPERATION ::= {
  ARGUMENT      SummarizeArgument
  RESULT        SummarizeResult
  ERRORS        {attribute-error | invalid-parameters-error | range-error |
                 security-error | sequence-number-error | service-error,
                 ... -- extensions de 1994 --,
                 entry-class-error | ms-extension-error}
  CODE          op-summarize }

```

NOTE – On trouvera à l'Annexe I un exemple d'opération de récapitulation.

8.2.1.1 Argument de récapitulation

```

SummarizeArgument ::= SET {
  entry-class    [0] EntryClass DEFAULT delivery,
  selector       [1] Selector,
  summary-requests [2] SEQUENCE SIZE (1..ub-summaries) OF ATTRIBUTE.&id ({AttributeTable})
                 OPTIONAL -- absent si aucune récapitulation n'est demandée --,
                 -- extensions de 1994 --
  summarize-extensions [3] MSExtensions OPTIONAL }

```

Les paramètres de l'argument de récapitulation **summarize-argument** ont la signification suivante:

- entry-class** (classe d'entrée) (optionnel): spécifie la classe d'entrées traitée par l'opération abstraite; voir 6.3.1;
- selector** (sélecteur) (obligatoire): ensemble de critères de sélection pour déterminer les entrées à récapituler; voir 8.1.3;
- summary-requests** (demandes de récapitulations) (optionnel): séquence de types d'attributs pour lesquels un récapitulatif est demandé. Ce paramètre n'est présent que si un récapitulatif est demandé;
- summarize-extensions** (extensions de récapitulation) (optionnel): ce paramètre permettra d'apporter ultérieurement à l'argument de récapitulation des extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.1.2 Résultat de récapitulation

Si la demande aboutit, le résultat de récapitulation **summarize-result** sera renvoyé.

```

SummarizeResult ::= SET {
  next           [0] SequenceNumber OPTIONAL,
  count          [1] INTEGER (0..ub-messages) -- décompte des entrées sélectionnées --,
  span           [2] Span OPTIONAL -- intervalle des entrées sélectionnées,
                 -- omis si la composante count vaut zéro --,
  summaries      [3] SEQUENCE SIZE (1..ub-summaries) OF Summary OPTIONAL,
                 -- extension de 1994 --
  summarize-result-extensions [4] MSExtensions OPTIONAL }

```

Les paramètres du résultat de récapitulation **summarize-result** ont la signification suivante:

- next** (suivant) (conditionnel): ce paramètre est renvoyé dans le cas où le nombre d'entrées sélectionnées aurait été plus grand si ce n'était la limite spécifiée dans le sélecteur. Ce paramètre contient le numéro de séquence de l'entrée suivante qui aurait alors été sélectionnée, conformément à l'ordre croissant ou décroissant de l'intervalle qui détermine le sens de la recherche (voir 8.1.3);
- count** (décompte) (obligatoire): entier donnant le décompte des entrées répondant aux critères de sélection;
- span** (portée) (conditionnel): ce paramètre contient le plus petit et le plus grand numéro de séquence des entrées répondant aux critères de sélection. Il est absent s'il n'existe pas de telles entrées.

```

Span ::= SEQUENCE {
  lowest [0] SequenceNumber,
  highest [1] SequenceNumber }

```

Les composantes du paramètre **span** (portée) ont la signification suivante:

- 1) **lowest** (inférieur) (obligatoire): identifie l'entrée qui possède le plus petit numéro de séquence et qui répond aux critères de sélection;
- 2) **highest** (supérieur) (obligatoire): identifie l'entrée qui possède le plus grand numéro de séquence et qui répond aux critères de sélection;

NOTE – Les valeurs des composantes **lowest** (inférieur) et **highest** (supérieur) ne sont pas affectées par l'ordre croissant ou décroissant de l'intervalle; voir 8.1.1.

- d) **summaries** (récapitulatifs) (conditionnel): un récapitulatif **summary** est renvoyé par demande de récapitulatif. Les **récapitulatifs** sont renvoyés dans l'ordre dans lequel ils ont été demandés.

```
Summary ::= SET {
    absent [0] INTEGER (1..ub-messages) OPTIONAL -- décompte des entrées dépourvues de
        l'attribut recherché --,
    present [1] SET SIZE (1..ub-attribute-values) OF -- une séquence pour chaque valeur d'attribut
        présente --
        SEQUENCE {
            type ATTRIBUTE.&id ({AttributeTable}),
            value ATTRIBUTE.&Type ({AttributeTable} {@.type}),
            count INTEGER (1..ub-messages) OPTIONAL }
}
```

Les composantes du paramètre **summary** (récapitulation) ont la signification suivante:

- 1) **absent** (conditionnel): décompte des entrées qui ne contiennent pas d'attribut du type spécifié dans la demande. Cette composante est omise s'il n'existe pas d'entrées de ce type;
- 2) **present** (présent) (conditionnel): récapitulation des entrées contenant un attribut du type spécifié, ventilé selon les valeurs d'attributs effectivement présentes. Cette composante est omise s'il n'existe pas d'entrées de ce type.

Les composantes de la composante **present** (présent) ont la signification suivante:

- i) **type** (obligatoire): type de l'attribut;
 - ii) **value** (valeur) (obligatoire): valeur d'attribut pour laquelle le décompte est donné;
 - iii) **count** (décompte) (obligatoire): décompte des entrées auxquelles est associée cette valeur d'attribut;
- e) **summarize-result-extensions** (extensions du résultat de récapitulation) (conditionnel): ce paramètre permettra d'apporter au résultat de récapitulation de futures extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.1.3 Erreurs abstraites de récapitulation

Si la demande échoue, une des erreurs abstraites énumérées sera signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9. L'erreur abstraite d'attribut doit être signalée à la fois si un filtre a été mal spécifié, et si le paramètre **summary-requests** (demandes de récapitulations) spécifie un type d'attribut qui n'est pas pris en charge par la mémoire de messages, ou si l'utilisateur MS n'y est pas abonné.

8.2.2 Opération abstraite List (listage)

L'opération abstraite **List** (listage) est utilisée pour rechercher dans une classe donnée les entrées présentant un intérêt et pour renvoyer les informations sélectionnées extraites de ces entrées.

L'opération abstraite **List** n'aboutira que si le contexte de sécurité et la politique de sécurité en vigueur permettent l'accès à la classe d'entrées.

Les informations qui peuvent être sélectionnées pour les entrées d'une classe donnée peuvent être restreintes. Les Tableaux 2 et 3 indiquent les restrictions applicables aux attributs généraux des entrées des diverses classes.

```
list ABSTRACT-OPERATION ::= {
    ARGUMENT      ListArgument
    RESULT        ListResult
    ERRORS        {attribute-error | invalid-parameters-error | range-error |
        security-error | sequence-number-error | service-error,
        ... -- extension de 1994 --,
        entry-class-error | ms-extension-error}
    CODE          op-list }
```

8.2.2.1 Argument de listage

```
ListArgument ::= SET {
    entry-class           [0] EntryClass DEFAULT delivery,
    selector             [1] Selector,
    requested-attributes [3] EntryInformationSelection OPTIONAL,
                        -- extension de 1994 --
    list-extensions      [4] MSExtensions OPTIONAL }
```

Les paramètres de l'argument de listage **list-argument** ont la signification suivante:

- entry-class** (classe d'entrée) (optionnel): spécifie la classe d'entrées traitée par l'opération abstraite; voir 6.3.1;
- selector** (sélecteur) (obligatoire): ensemble de critères de sélection pour déterminer les entrées à renvoyer; voir 8.1.3;
- requested-attributes** (attributs demandés) (optionnel): indique les informations des entrées sélectionnées qui devront être renvoyées dans le résultat; voir 8.1.4. Si ce paramètre est absent, l'ensemble consigné des valeurs par défaut des attributs de listage d'agent utilisateur est utilisé. Si aucune consignation d'agent utilisateur n'a été spécifiée lors de l'établissement de l'association abstraite, ou si aucune valeur par défaut d'attribut de listage d'agent utilisateur n'est consignée pour la consignation d'agent utilisateur active en cours, les valeurs générales (qui ne sont pas propres à l'agent utilisateur) par défaut des attributs de listage sont utilisées; voir 8.2.5.1 c) et g);
- list-extensions** (extensions de listage) (optionnel): ce paramètre permettra d'apporter à l'argument de listage de futures extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.2.2 Résultat de listage

Si la demande aboutit, le résultat de listage **list-result** sera renvoyé.

```
ListResult ::= SET {
    next                 [0] SequenceNumber OPTIONAL,
    requested            [1] SEQUENCE SIZE (1..ub-messages) OF EntryInformation OPTIONAL
                        -- omis si aucune entrée n'est trouvée --,
                        -- extension de 1994 --
    list-result-extensions [2] MSExtensions OPTIONAL }
```

Les paramètres du résultat de listage **list-result** ont la signification suivante:

- next** (suivant) (conditionnel): ce paramètre est renvoyé lorsque le nombre d'entrées sélectionnées aurait été plus grand si ce n'était l'absence de la limite spécifiée dans le sélecteur. Ce paramètre contient le numéro de séquence de l'entrée suivante qui aurait été sélectionnée, d'après l'ordre croissant ou décroissant de l'intervalle, qui détermine le sens de la recherche (voir 8.1.3);
- requested** (demandé) (conditionnel): ce paramètre véhicule les informations demandées (voir 8.1.5) extraites de chaque entrée sélectionnée. Les éléments d'information de l'entrée apparaissent dans l'ordre croissant ou décroissant de l'intervalle (voir 8.1.3). Le paramètre est absent si aucune entrée n'est sélectionnée;
- list-result-extensions** (extensions du résultat de listage) (conditionnel): ce paramètre permettra ultérieurement d'apporter au résultat de listage des extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.2.3 Erreurs abstraites de listage

Si la demande échoue, une des erreurs abstraites énumérées sera signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9.

8.2.3 Opération abstraite Fetch (recherche)

L'opération abstraite **Fetch** (recherche) sert à renvoyer des informations sélectionnées extraites d'une entrée particulière d'une classe donnée. Elle peut aussi être utilisée pour renvoyer des informations sélectionnées extraites de la première entrée intéressante; dans ce cas, les numéros de séquence des autres entrées sélectionnées sont également renvoyés. L'opération abstraite Fetch (recherche) n'aboutira que si le contexte de sécurité et la politique de sécurité en vigueur permettent l'accès à la classe d'entrées.

Les informations provenant d'une entrée peuvent être recherchées plusieurs fois jusqu'à ce que l'entrée ait été explicitement supprimée par l'opération abstraite Delete (suppression).

```

fetch ABSTRACT-OPERATION ::= {
  ARGUMENT           FetchArgument
  RESULT            FetchResult
  ERRORS           {attribute-error | fetch-restriction-error | invalid-parameters-error |
                    range-error | security-error | sequence-number-error | service-error,
                    ... -- extensions de 1994 --,
                    entry-class-error | ms-extension-error}
  CODE             op-fetch }

```

8.2.3.1 Argument de recherche

```

FetchArgument ::= SET {
  entry-class       [0] EntryClass DEFAULT delivery,
  item             CHOICE {
                    search       [1] Selector,
                    precise     [2] SequenceNumber},
  requested-attributes [3] EntryInformationSelection OPTIONAL,
                    -- extension de 1994 --
  fetch-extensions [4] MSExtensions OPTIONAL }

```

Les paramètres de l'argument de recherche **fetch-argument** ont la signification suivante:

- a) **entry-class** (classe d'entrée) (optionnel): ce paramètre spécifie la classe d'entrées traitée par l'opération abstraite; voir 6.3.1;
- b) **item** (élément) (obligatoire): une des composantes décrites ci-dessous sera spécifiée pour déterminer l'entrée à rechercher:
 - 1) **search** (recherche) (conditionnel): sélecteur spécifiant un ensemble d'entrées. Parmi les entrées sélectionnées (s'il en existe), l'entrée recherchée sera celle qui porte le numéro de séquence le plus élevé si l'intervalle est défini dans l'ordre décroissant, et celle qui porte le numéro de séquence le plus petit sinon; voir 8.1.3;
 - 2) **precise** (arrêt) (conditionnel): numéro de séquence de l'entrée à rechercher; voir 6.3.2;
- c) **requested-attributes** (attributs demandés) (optionnel): ce paramètre indique les informations à extraire de l'entrée sélectionnée et à renvoyer dans le résultat; voir 8.1.4. Si ce paramètre est absent, l'ensemble consigné des valeurs par défaut des attributs de recherche d'agent utilisateur est utilisé. Si aucune consignation d'agent utilisateur n'a été spécifiée lors de l'établissement de l'opération abstraite, ou si la consignation d'agent utilisateur active ne spécifie pas de valeur par défaut d'attribut de recherche d'agent utilisateur, les valeurs générales (non propres à l'agent utilisateur) par défaut des attributs de recherche sont utilisées; voir 8.2.5.1 d) et g);
- d) **fetch-extensions** (extensions de recherche) (optionnel): ce paramètre permettra ultérieurement d'apporter à l'argument de recherche des extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.3.2 Résultat de recherche

Si la demande aboutit, le résultat de recherche **fetch-result** sera renvoyé.

```

FetchResult ::= SET {
  entry-information [0] EntryInformation OPTIONAL -- si une entrée a été sélectionnée --,
  list             [1] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
  next            [2] SequenceNumber OPTIONAL,
                    -- extension de 1994 --
  fetch-result-extensions [3] MSExtensions OPTIONAL }

```

Les paramètres du résultat de recherche **fetch-result** ont la signification suivante:

- a) **entry-information** (information d'entrée) (conditionnel): ensemble des attributs demandés présents dans l'entrée sélectionnée; voir 8.1.5. Ce paramètre est absent lorsqu'une recherche a été effectuée et qu'aucune entrée n'a été sélectionnée;

- b) **list** (liste) (conditionnel): ce paramètre est renvoyé lorsque plusieurs entrées ont répondu au sélecteur de recherche. La liste donne les numéros de séquence de ces entrées dans l'ordre croissant ou décroissant défini par l'intervalle (voir 8.1.3);
- c) **next** (entrée suivante) (conditionnel): ce paramètre est renvoyé lorsque le nombre d'entrées sélectionnées aurait été plus grand si ce n'était la limite spécifiée dans le sélecteur. Ce paramètre comprend le numéro de séquence de l'entrée suivante qui aurait été sélectionnée, d'après l'ordre croissant ou décroissant de l'intervalle qui détermine le sens de recherche;
- d) **fetch-result-extensions** (extensions du résultat de recherche) (conditionnel): ce paramètre permettra d'apporter ultérieurement au résultat de recherche des extensions générales et propres au contenu. Aucune extension n'est définie dans la présente Définition de service.

8.2.3.3 Erreurs abstraites de recherche

Si la demande échoue, une des erreurs abstraites énumérées sera signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9.

8.2.4 Opération abstraite Delete (suppression)

L'opération abstraite **Delete** (suppression) est utilisée pour supprimer les entrées sélectionnées d'une classe spécifiée. Une entrée principale et toutes les entrées filles qui lui sont rattachées ne peuvent être supprimées qu'ensemble. Cette opération est effectuée en spécifiant uniquement l'entrée principale comme argument. L'opération abstraite Delete n'aboutira que si la classe d'entrées autorise la suppression, et cela est autorisé par le contexte de sécurité et par la politique de sécurité en vigueur.

Certaines entrées ne sont pas soumises à la suppression. Une tentative de suppression d'une entrée fille se traduira par une erreur de suppression de type *child-entry-specified* (entrée fille spécifiée). Une tentative de suppression d'une entrée de la classe Delivery (remise) dont l'état d'extraction est à *nouveau* (voir 6.3.8) se traduira par une erreur de suppression de type *new-entry-specified* (nouvelle entrée spécifiée); l'état d'extraction ne doit pas être pris en considération lorsque des entrées appartenant à des classes d'entrée autres que Delivery (remise) sont supprimées. L'état d'extraction de toute entrée fille associée à une entrée principale ne doit pas être pris en considération lorsqu'on applique l'opération abstraite Delete (suppression) à l'entrée principale. Une tentative de suppression d'une entrée de la classe Message-log (journal de messagerie) pour laquelle une entrée correspondante existe dans la classe d'entrées Stored-message (message enregistré) se traduira par une erreur de suppression de type *stored-message-exists* (message enregistré existant).

Comme option à l'abonnement, la suppression d'entrées d'une ou plusieurs classes d'entrée quelconques peut être restreinte ou interdite, et peut entraîner une erreur de suppression de type *entry-class-restriction* (restriction de classe d'entrée). D'autres restrictions peuvent être définies pour les entrées d'un type de contenu donné comme cela est indiqué dans la Spécification qui définit ce type de contenu. Une tentative de transgression d'une telle restriction de suppression entraîne la création d'une erreur de suppression de type *delete-restriction-problem* (problème concernant les restrictions de suppression).

NOTE – Les réalisations peuvent choisir d'interdire la suppression d'entrées de la classe Message-log (journal de messagerie) à l'exception de l'entrée la plus ancienne, ou d'une séquence contiguë qui comprend la plus ancienne.

```
delete ABSTRACT-OPERATION ::= {
  ARGUMENT DeleteArgument
  RESULT DeleteResult
  ERRORS {deleteError | invalid-parameters-error | range-error | security-error |
sequence-number-error | service-error,
... -- extensions de 1994 --,
entry-class-error | ms-extension-error}
  CODE op-delete }
```

8.2.4.1 Argument de suppression

```
DeleteArgument ::= SET {
  entry-class [0] EntryClass DEFAULT delivery,
  items CHOICE {
    selector [1] Selector,
    sequence-numbers [2] SET SIZE (1..ub-messages) OF SequenceNumber },
  -- extension de 1994 --
  delete-extensions [3] MSExtensions OPTIONAL }
```

Les paramètres de l'argument de suppression **delete-argument** ont la signification suivante:

- a) **entry-class** (classe d'entrée) (optionnel): spécifie la classe d'entrées appelée par l'opération abstraite; voir 6.3.1;
- b) **items** (éléments) (obligatoire): l'une des composantes décrites ci-dessous doit être spécifiée afin de déterminer les entrées à supprimer:
 - 1) **selector** (sélecteur) (conditionnel): voir 8.1.3;
 - 2) **sequence-numbers** (numéros de séquence) (conditionnel): liste non ordonnée de numéros de séquence; voir 6.3.2;
- c) **delete-extensions** (extensions de suppression) (optionnel): ce paramètre permet de futures extensions générales et propres au contenu concernant l'argument de suppression. Aucune extension n'est définie dans la présente Définition de service.

NOTE – Le type de contenu Interpersonal Messaging (messagerie de personne à personne) utilise les extensions de suppression définies au 19.5.3 de la Rec. UIT-T X.420 | ISO/CEI 10021-7.

8.2.4.2 Résultat de suppression

Si la demande aboutit, le résultat de suppression **delete-result** sera renvoyé.

```

DeleteResult ::= CHOICE {
  delete-result-88      NULL,
                        -- extension de 1994 --
  delete-result-94      SET {
    entries-deleted      [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
    delete-result-extensions [1] MExtensions OPTIONAL } }

```

Les paramètres du résultat de suppression **delete-result** ont la signification suivante:

- a) **delete-result-88** (résultat de suppression 88) (conditionnel): ce paramètre est renvoyé en cas d'utilisation d'un contexte d'application 1988;
- b) **delete-result-94** (résultat de suppression 94) (conditionnel): ce paramètre est renvoyé en cas d'utilisation d'un contexte d'application 1994. Il comprend les composantes suivantes:
 - 1) **entries-deleted** (entrées supprimées) (conditionnel): cette composante identifie les entrées supprimées. Elle est présente si la composante selector (sélecteur) était présente dans l'argument de suppression, et si au moins une entrée a été supprimée. Elle est absente sinon;
 - 2) **delete-result-extensions** (extensions du résultat de suppression) (conditionnel): cette composante permet de futures extensions générales et propres au contenu concernant le résultat de suppression. Aucune extension n'est définie dans la présente Définition de service.

8.2.4.3 Erreurs abstraites de suppression

Si la demande échoue, une des erreurs abstraites énumérées sera signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9.

8.2.5 Opération abstraite Register-MS (consignation MS)

L'opération abstraite **Register-MS** (consignation MS) sert à consigner et à déconsigner diverses informations avec la mémoire de messages, et d'extraire des informations consignées de la mémoire de messages:

- a) actions automatiques;
- b) valeurs par défaut des types d'attributs demandés pour les opérations de listage et de recherche;
- c) pouvoirs;
- d) étiquettes de sécurité de l'utilisateur;
- e) consignations d'agent utilisateur;
- f) valeurs par défaut de dépôt;
- g) noms de groupes de messages.

Lorsqu'une mémoire de messages prend en charge à la fois un contexte d'application 1988 et 1994 pour un certain utilisateur MS, les consignations faites en utilisant l'un des contextes d'application s'appliqueront lors de l'utilisation de l'autre contexte d'application.

NOTE – Par exemple, les valeurs par défaut générales de dépôt consignées lors de l'utilisation d'un contexte d'application 1994 s'appliqueront aux dépôts invoqués dans une association abstraite établie en utilisant un contexte d'application 1988.

```

register-MS ABSTRACT-OPERATION ::= {
  ARGUMENT      Register-MSArgument
  RESULT        Register-MSResult
  ERRORS        {attribute-error | auto-action-request-error | invalid-parameters-error |
                 security-error | service-error | old-credentials-incorrectly-specified |
                 new-credentials-unacceptable,
                 ... -- extensions de 1994 --,
                 message-group-error | ms-extension-error | register-ms-error}
  CODE          op-register-ms }

```

8.2.5.1 Argument de consignation MS

```

Register-MSArgument ::= SET {
  auto-action-registrations [0] SET SIZE (1..ub-auto-registrations) OF AutoActionRegistration
                           OPTIONAL,
  auto-action-deregistrations [1] SET SIZE (1..ub-auto-registrations) OF AutoActionDeregistration
                              OPTIONAL,
  list-attribute-defaults [2] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id
                           (AttributeTable) OPTIONAL,
  fetch-attribute-defaults [3] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id
                           (AttributeTable) OPTIONAL,
  change-credentials [4] SEQUENCE {
    old-credentials [0] Credentials,
    new-credentials [1] Credentials } OPTIONAL,
  user-security-labels [5] SET SIZE (1..ub-labels-and-redirections) OF SecurityLabel OPTIONAL,
  -- extension de 1994 --
  ua-registrations [6] SET SIZE (1..ub-ua-registrations) OF UARegistration OPTIONAL,
  submission-defaults [7] MSSubmissionOptions OPTIONAL,
  message-group-registrations [8] MessageGroupRegistrations OPTIONAL,
  registration-status-request [9] RegistrationTypes OPTIONAL,
  register-ms-extensions [10] MSExtensions OPTIONAL }

```

Les paramètres de l'argument de consignation MS **register-MS-argument** ont la signification suivante:

- a) **auto-action-registrations** (consignations d'actions automatiques) (optionnel): ensemble de **consignations d'actions automatiques** (voir 6.5.2), une consignation pour chaque action automatique à consigner. Le nouveau paramètre de consignation **registration-parameter** d'actions automatiques remplace toute action automatique précédemment consignée (le cas échéant) comportant cet identificateur de consignation et ce type d'action automatique.

En l'absence du paramètre **auto-action-registrations**, aucune nouvelle action automatique n'est consignée;

- b) **auto-action-deregistrations** (déconsignations d'actions automatiques) (optionnel): ensemble de **déconsignations d'actions automatiques**, une déconsignation pour chaque action automatique à déconsigner. Toute action automatique consignée comportant un identificateur de consignation **registration-identifier** et un type d'action automatique **auto-action-type** conformes à ceux qui sont spécifiés dans la déconsignation d'action automatique est déconsignée.

```

AutoActionDeregistration ::= SEQUENCE {
  auto-action-type      AUTO-ACTION.&id (AutoActionTable),
  registration-identifier [0] INTEGER (1..ub-per-auto-action) DEFAULT 1 }

```

En l'absence du paramètre **auto-action-deregistrations**, l'ensemble existant d'actions automatiques consignées reste inchangé, sauf s'il a été mis à jour par le paramètre **auto-action-registrations** (consignations d'actions automatiques);

- c) **list-attribute-defaults** (valeurs par défaut des attributs de listage) (optionnel): spécifie les types d'attributs qui seront renvoyés dans toute opération abstraite **List** (listage) invoquée ultérieurement si le paramètre **requested-attributes** (attributs demandés) est absent, et si soit aucune consignation d'agent utilisateur n'a été spécifiée lors de l'établissement de l'association abstraite, soit la consignation d'agent utilisateur active ne spécifie pas de valeur par défaut pour les attributs de listage d'agent utilisateur.

En l'absence du paramètre **list-attribute-defaults** (valeurs par défaut des attributs de listage), aucun changement n'est apporté à la valeur par défaut consignée (le cas échéant). Les valeurs par défaut des attributs de listage forment l'ensemble vide jusqu'à ce que cet ensemble soit modifié explicitement par l'utilisateur MS via l'opération abstraite **Register-MS** (consignation MS);

NOTE 1 – Dans les versions de la présente Définition de service publiées avant 1994, l'ensemble possédait une borne inférieure de un.

- d) **fetch-attribute-defaults** (valeurs par défaut des attributs de recherche) (optionnel): spécifie les types d'attributs qui seront renvoyés dans toute opération abstraite Fetch (recherche) invoquée ultérieurement si le paramètre requested-attributes (attributs demandés) est absent, et si soit aucune consignation d'agent utilisateur n'a été spécifiée lors de l'établissement de l'association abstraite, soit la consignation d'agent utilisateur active ne spécifie pas de valeur par défaut pour les attributs de recherche d'agent utilisateur.

En l'absence du paramètre fetch-attribute-defaults (valeurs par défaut des attributs de recherche), aucun changement n'est apporté à la valeur par défaut consignée (le cas échéant). Les valeurs par défaut des attributs de recherche forment l'ensemble vide jusqu'à ce que cet ensemble soit modifié explicitement par l'utilisateur MS via l'opération abstraite Register-MS (consignation MS);

NOTE 2 – Dans les versions de la présente Définition de service publiées avant 1994, l'ensemble possédait une borne inférieure de un.

- e) **change-credentials** (modification des pouvoirs) (optionnel): spécifie une modification des pouvoirs utilisés dans le rattachement MS pour authentifier l'identité de l'utilisateur MS à la mémoire de messages.

Les anciens pouvoirs **old-credentials** sont les pouvoirs actuels de l'utilisateur MS; les nouveaux pouvoirs **new-credentials** sont les pouvoirs proposés pour une utilisation future dans un rattachement MS.

En l'absence de ce paramètre, les pouvoirs précédemment consignés restent en l'état.

Les pouvoirs de l'utilisateur MS peuvent différer en valeur des pouvoirs du demandeur **initiator-credentials** spécifiés au 8.1.1.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Les pouvoirs utilisés entre l'utilisateur MTS (la mémoire de messages) et le système MTS sont maintenus de façon séparée et peuvent être modifiés au moyen de l'opération abstraite Register (consignation) de l'accès d'administration;

- f) **user-security-labels** (étiquettes de sécurité de l'utilisateur) (optionnel): ce paramètre contient la ou les étiquettes de sécurité **security-label** de l'utilisateur MS, si elles doivent être modifiées. Le paramètre user-security-labels est défini au 8.4.1.1.1.7 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

En l'absence de ce paramètre, les étiquettes de sécurité de l'utilisateur restent inchangées;

NOTE 3 – Selon certaines politiques de sécurité, les étiquettes de sécurité de l'utilisateur ne peuvent être ainsi modifiées que si une liaison sécurisée est utilisée. D'autres moyens locaux pour modifier les étiquettes de sécurité de l'utilisateur de façon sécurisée peuvent être fournis.

- g) **UA-registrations** (consignations d'agent utilisateur) (optionnel): ce paramètre spécifie une ou plusieurs modifications à l'ensemble des consignations d'agent utilisateur. Si l'argument de rattachement MS de la présente association abstraite ou d'une association abstraite établie ultérieurement identifie une consignation d'agent utilisateur qui a été consignée à l'aide de ce paramètre, les valeurs par défaut spécifiées ci-dessous, associées à cette consignation, s'appliqueront pendant la durée de l'association abstraite; voir 7.1.1 f).

NOTE 4 – A diverses occasions, un utilisateur peut utiliser différents agents utilisateur pour communiquer avec la mémoire de messages.

```

UARegistration ::= SET {
    ua-registration-identifieur      [0] RegistrationIdentifier,
    ua-list-attribute-defaults       [1] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id
                                     (AttributeTable) OPTIONAL,
    ua-fetch-attribute-defaults      [2] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id
                                     (AttributeTable) OPTIONAL,
    ua-submission-defaults           [3] MSSubmissionOptions OPTIONAL,
    content-specific-defaults        [4] MSExtensions OPTIONAL }
    
```

Les composantes du paramètre **UA-registrations** (consignation d'agent utilisateur) sont les suivantes:

- 1) **UA-registration-identifieur** (identificateur de consignation d'agent utilisateur) (obligatoire): il s'agit d'un identificateur pour cette consignation d'agent utilisateur. Si la même valeur d'identification de consignation d'agent utilisateur a été spécifiée dans une précédente invocation de cette opération abstraite, les composantes présentes dans la nouvelle consignation remplacent les composantes correspondantes de la précédente consignation; sinon une nouvelle consignation d'agent utilisateur est créée. Si la valeur d'identification de consignation d'agent utilisateur est conforme à celle qui est présente de façon facultative dans l'argument de rattachement MS qui a établi l'association abstraite en cours, la nouvelle consignation d'agent utilisateur prend effet immédiatement.

Si l'identificateur de consignation d'agent utilisateur est la seule composante présente dans une consignation d'agent utilisateur, aucune consignation d'agent utilisateur n'est enregistrée et toute consignation d'agent utilisateur existante comportant cet identificateur de consignation d'agent utilisateur est supprimée;

NOTE 5 – Il est probable qu'une consignation d'agent utilisateur contienne des valeurs influencées à la fois par la conception de l'agent utilisateur et par les options commandées par l'utilisateur. Il peut être souhaitable que l'identificateur de consignation d'agent utilisateur soit rendu configurable par l'utilisateur plutôt que fixé dans la réalisation de l'agent utilisateur.

- 2) **UA-list-attribute-defaults** (valeurs par défaut des attributs de listage de l'agent utilisateur) (optionnel): cette composante spécifie les types d'attributs qui seront renvoyés dans une opération abstraite List (listage) exécutée ultérieurement si le paramètre entry-information-selection (sélection d'informations d'entrée) est absent. En l'absence de cette composante, aucun changement n'est apporté aux valeurs par défaut des attributs de listage de l'agent utilisateur définies pour cette consignation d'agent utilisateur (le cas échéant);
- 3) **UA-fetch-attribute-defaults** (valeurs par défaut des attributs de recherche de l'agent utilisateur) (optionnel): cette composante spécifie les types d'attributs qui seront renvoyés dans une opération abstraite Fetch (recherche) exécutée ultérieurement si le paramètre entry-information-selection (sélection d'informations d'entrée) est absent. En l'absence de cette composante, aucun changement n'est apporté aux valeurs par défaut des attributs de recherche de l'agent utilisateur définies pour cette consignation d'agent utilisateur (le cas échéant);
- 4) **UA-submission-defaults** (valeurs par défaut de dépôt de l'agent utilisateur) (optionnel): cette composante spécifie le paramètre submission-options (options de dépôt) qui sera utilisé dans une opération abstraite MS-message-submission (dépôt de message MS) ou MS-probe-submission (dépôt de message d'essai MS) exécutée ultérieurement lorsque le paramètre submission-options (options de dépôt) est absent. La valeur *draft* (brouillon) n'est pas autorisée pour la composante object-entry-class (classe d'entrées de l'objet); voir 8.1.6. En l'absence de cette composante, aucun changement n'est apporté aux valeurs par défaut de dépôt de l'agent utilisateur définies pour cette consignation d'agent utilisateur (le cas échéant);
- 5) **content-specific-defaults** (valeurs par défaut propres au contenu) (optionnel): cette composante spécifie les consignations propres à un type de contenu particulier. La définition d'une valeur par défaut propre au contenu apparaît dans la Spécification pour le type de contenu concerné. En l'absence de cette composante, aucun changement n'est apporté aux valeurs par défaut propres au contenu définies pour cette consignation d'agent utilisateur (le cas échéant).

Toute consignation d'agent utilisateur existante omise dans le paramètre UA-registrations (consignations d'agent utilisateur) reste inchangée. En l'absence du paramètre UA-registrations, aucun changement n'est apporté aux consignations existantes;

- h) **submission-defaults** (valeurs par défaut de dépôt) (optionnel): si l'argument submission-options (options de dépôt) de l'opération abstraite MS-message-submission (dépôt de message MS) ou MS-probe-submission (dépôt de message d'essai MS) est absent lorsque ces opérations abstraites sont invoquées, et qu'aucune consignation d'agent utilisateur n'est spécifiée lorsque l'association abstraite est établie (ou que cette consignation d'agent utilisateur n'a pas spécifié de valeur par défaut de dépôt de l'agent utilisateur), ou lorsque la mémoire de messages elle-même exécute un dépôt tout en traitant une action automatique, ce paramètre fournit des valeurs par défaut pour l'argument submission-options. L'état initial du paramètre, qui est décrit au 8.1.6, est que toutes ses composantes sont absentes et sont positionnées à leurs valeurs par défaut. Ces valeurs restent les mêmes jusqu'à ce qu'elles soient modifiées par une invocation de l'opération abstraite Register-MS (consignation MS). La valeur *draft* (brouillon) n'est pas autorisée pour la composante object-entry-class (classe d'entrées de l'objet). En l'absence de ce paramètre, la valeur du paramètre submission-defaults reste inchangée;
- i) **message-group-registrations** (consignations de groupes de messages) (optionnel): ce paramètre spécifie les modifications de l'ensemble des noms de groupes de messages consignés par l'utilisateur MS qui peut être utilisé pour la classification et l'organisation des entrées de la mémoire de messages; voir 6.4. Une tentative d'ajout d'une valeur à l'attribut message-group-name (nom de groupe de messages) d'une entrée se traduira par une erreur sauf si la valeur spécifiée a été précédemment consignée à l'aide de ce paramètre.

```

MessageGroupRegistrations ::= SEQUENCE SIZE (1..ub-default-registrations) OF CHOICE {
    register-group           [0] MessageGroupNameAndDescriptor,
    deregister-group        [1] MessageGroupName,
    change-descriptors      [2] MessageGroupNameAndDescriptor }

```

```

MessageGroupNameAndDescriptor ::= SET {
    message-group-name [0] MessageGroupName,
    message-group-descriptor [1] GeneralString (SIZE (1..ub-group-descriptor-length)) OPTIONAL }

```

Les composantes du paramètre **message-group-registrations** (consignations de groupes de messages) sont les suivantes:

- 1) **register-group** (consignation de groupe) (optionnel): le nom de groupe de messages spécifié est ajouté à l'ensemble des groupes de messages consignés. Un nom de groupe de messages reste consigné jusqu'à ce qu'il soit explicitement déconsigné dans une opération abstraite Register-MS (consignation MS) ultérieure. Une tentative de consignation d'un nom de groupe de messages qui est déjà consigné se traduit par une erreur. Lorsqu'un nom de groupe de messages spécifié contient plusieurs parties de nom de groupe, une erreur en résulte sauf si une consignation existe actuellement pour ce nom moins sa partie de nom de groupe finale.

NOTE 6 – Par conséquent, chaque nom de groupe de messages qui contient plusieurs parties de nom de groupe a un nom de groupe de messages père au-dessus de lui dans la hiérarchie imaginaire des noms de groupes de messages.

Le descripteur de groupe de messages **message-group-descriptor** est une composante optionnelle qui fournit une description informelle de chaque groupe de messages consigné;

- 2) **deregister-group** (déconsignation de groupe) (optionnel): le nom de groupe de messages spécifié est retiré de l'ensemble des consignations de groupes de messages. Une erreur de groupe de messages est signalée si n'importe quel nom de groupe de messages fils (c'est-à-dire un nom formé en ajoutant une partie de nom de groupe au nom du groupe de messages spécifié) est actuellement consigné. Une erreur de groupe de messages est également signalée si le nom de groupe de messages spécifié n'est pas consigné, ou est utilisé, c'est-à-dire s'il est affecté à l'attribut **message-group-name** (nom de groupe de messages) de n'importe quelle entrée, ou est référencé dans une action automatique Auto-modify (modification automatique) consignée;
- 3) **change-descriptors** (modification des descripteurs) (optionnel): la valeur fournie dans le paramètre **message-group-descriptor** (descripteur de groupe de messages) remplace la valeur enregistrée pour le nom de groupe de messages spécifié. Une erreur de groupe de messages est signalée si un nom de groupe de messages spécifié n'est pas consigné.

Les composantes du paramètre **message-group-registrations** (consignations de groupes de messages) seront appliquées dans l'ordre présenté. En l'absence du paramètre **message-group-registrations**, aucun changement n'est apporté à l'ensemble consigné de noms de groupes de messages ni aux descripteurs;

- j) **registration-status-request** (demande d'état de consignation) (optionnel): ce paramètre sert à demander des informations à la mémoire de messages concernant les positionnements courants des valeurs par défaut et d'autres éléments consignés. Le résultat renvoyé reflète l'état des informations consignées après que tous les autres paramètres de l'argument de consignation MS ont été traités. Le paramètre contient plusieurs éléments dont chacun, s'il est fixé, demande l'état de consignation de la classe d'informations correspondante. En l'absence de ce paramètre, aucune information concernant l'état de consignation n'est demandée.

```

RegistrationTypes ::= SET {
    registrations [0] BIT STRING {
        auto-action-registrations (0),
        list-attribute-defaults (1),
        fetch-attribute-defaults (2),
        ua-registrations (3),
        submission-defaults (4),
        message-group-registrations (5) } OPTIONAL,
    extended-registrations [1] SET OF MS-EXTENSION.&id OPTIONAL,
    restrict-message-groups [2] MessageGroupsRestriction OPTIONAL }

```

Les composantes du paramètre **registration-types** (types de consignation) ont la signification suivante:

- 1) **registrations** (consignations) (optionnel): spécifie les éléments des informations consignées demandés. Un ou plusieurs des éléments: consignations d'action automatique, valeurs par défaut d'attribut de listage, valeurs par défaut d'attribut de recherche, consignations d'agent utilisateur, valeurs par défaut de dépôt et consignations de groupe de messages, pourront être demandés;
- 2) **extended-registration** (consignations étendues) (optionnel): spécifie les éléments des informations consignées étendues demandés;
- 3) **restrict-message-groups** (groupes de messages restreints) (optionnel): si le paramètre **message-group-registrations** (consignations de groupes de messages) est demandé, cette composante spécifie les restrictions sur les informations renvoyées.

```

MessageGroupsRestriction ::= SET {
  parent-group           [0] MessageGroupName OPTIONAL,
  immediate-descendants-only [1] BOOLEAN DEFAULT TRUE,
  omit-descriptors       [2] BOOLEAN DEFAULT TRUE }

```

Si un groupe père **parent-group** est spécifié, seuls les groupes de messages consignés qui sont les fils de ce groupe père seront renvoyés, c'est-à-dire les noms de groupes de messages qui sont formés en ajoutant une ou plusieurs parties de nom de groupe au nom du groupe père. Si la composante **immediate-descendants-only** (seuls les descendants immédiats) est *vraie*, seuls les noms de groupes de messages formés en ajoutant une seule partie de nom de groupe au nom du groupe père sont renvoyés; sinon tous les noms de groupes de messages descendants sont renvoyés. Si la composante **immediate-descendants-only** (seuls les descendants immédiats) est *vraie* et si le groupe père est absent, seuls les noms de groupes de messages qui contiennent une seule partie de nom de groupe sont renvoyés. Si la composante **omit-descriptors** (descripteurs omis) est *vraie*, les descripteurs de groupes de messages seront omis dans le résultat.

Si la composante **restrict-message-groups** (groupes de messages restreints) est absente [et que le paramètre **message-group-registrations** (consignations de groupes de messages) est demandé], tous les descripteurs et tous les noms de groupes de messages consignés sont renvoyés. La composante **restrict-message-groups** sera omise si le paramètre **message-group-registrations** n'est pas spécifié;

- k) **register-MS-extensions** (extensions de consignation MS) (optionnel): ce paramètre permet de futures extensions générales et propres au contenu concernant l'argument de consignation MS. Aucune extension n'est définie dans la présente Définition de service.

8.2.5.2 Résultat de consignation

Si la demande aboutit, le résultat de consignation MS **register-MS-result** sera renvoyé.

```

Register-MSResult ::= CHOICE {
  no-status-information    NULL,
  -- extension de 1994 --
  registered-information SET {
    auto-action-registrations [0] SET SIZE (1..ub-auto-registrations) OF AutoActionRegistration
                                OPTIONAL,
    list-attribute-defaults  [1] SET SIZE (1..ub-default-registrations) OF ATTRIBUTE.&id
                                (AttributeTable) OPTIONAL,
    fetch-attribute-defaults [2] SET SIZE (1..ub-default-registrations) OF ATTRIBUTE.&id
                                (AttributeTable) OPTIONAL,
    ua-registrations         [3] SET SIZE (1..ub-ua-registrations) OF UARegistration OPTIONAL,
    submission-defaults     [4] MSSubmissionOptions OPTIONAL,
    message-group-registrations [5] SET SIZE (1..ub-message-groups) OF
                                MessageGroupNameAndDescriptor OPTIONAL,
    register-ms-result-extensions [6] MSExtensions OPTIONAL } }

```

Les paramètres du résultat de consignation MS **register-MS-result** ont la signification suivante:

- a) **no-status-information** (pas d'information d'état) (conditionnel): présent si le paramètre **registration-status-request** (demande d'état de consignation) était absent dans l'argument de consignation MS;
- b) **registered-information** (information consignée) (conditionnel): ce paramètre contient les informations demandées dans le paramètre **registration-status-request** (demande d'état de consignation) de l'argument de consignation MS. Chaque composante du paramètre **registered-information** correspond à la composante de nom similaire de l'argument de consignation. La présence d'un élément d'information dépend du positionnement de la composante correspondante du paramètre **registration-status-request**, et de la mise à disposition d'une valeur pour cet élément. Les informations acheminées dans le résultat reflètent les valeurs des informations consignées à la suite du traitement de toute consignation ou déconsignation demandée dans l'argument de consignation MS. La composante **register-MS-result-extensions** (extensions du résultat de consignation MS) permet de futures extensions générales et propres au contenu concernant le résultat de consignation MS. Aucune extension n'est définie dans la présente Définition de service.

8.2.5.3 Erreurs abstraites de consignation MS

Si la demande échoue, une des erreurs abstraites énumérées doit être signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9. En plus des erreurs abstraites qui interviennent directement [par exemple à partir d'un paramètre `change-credentials` (modification des pouvoirs) non valable], la mémoire de messages créera une erreur abstraite lorsqu'une consignation affectant l'exécution ultérieure des opérations abstraites `List` (listage) ou `Fetch` (recherche), ou d'une action automatique entraînerait l'échec de cette opération abstraite ou de cette action automatique.

NOTE – Cela se produit, par exemple, lorsque le paramètre `list-attribute-defaults` (valeurs par défaut des attributs de listage) comprend un type d'attribut non disponible, ou lorsqu'un paramètre `auto-action-registration` (consignation d'action automatique) contient un filtre qui fait référence à un type d'attribut non disponible.

8.2.6 Opération abstraite Alert (avertissement)

L'opération abstraite **Alert** (avertissement) permet à la mémoire de messages d'informer l'utilisateur MS de la remise d'un message ou d'un rapport, dont les attributs sont conformes aux critères de sélection de l'un des paramètres de consignation des avertissements automatiques (voir 13.1), précédemment fournis par abonnement ou à l'aide de l'opération abstraite `Register-MS` (consignation MS).

L'opération abstraite **Alert** peut être invoquée uniquement sur une association abstraite existante établie par l'utilisateur MS, et uniquement après la création d'entrées, une fois l'association abstraite établie.

La présence d'entrées qui ont été créées entre associations abstraites et qui sont conformes aux critères de sélection consignés sera indiquée dans le résultat de l'opération abstraite `MS-bind` (rattachement MS) qui établit l'association abstraite suivante. Aucune opération abstraite **Alert** ne sera invoquée pour ces entrées. Voir 7.1.2 d).

L'opération abstraite **Alert** n'aboutira que si l'accès à la classe d'entrées `Delivery` (remise) est autorisé conformément au contexte de sécurité et à la politique de sécurité en vigueur.

```

alert ABSTRACT-OPERATION ::= {
    ARGUMENT      AlertArgument
    RESULT        AlertResult
    ERRORS        {security-error}
    CODE          op-alert }

```

8.2.6.1 Argument d'avertissement

```

AlertArgument ::= SET {
    alert-registration-identifiant [0] INTEGER (1..ub-auto-actions),
    new-entry                      [2] EntryInformation OPTIONAL }

```

Les paramètres de l'argument d'avertissement **alert-argument** ont la signification suivante:

- alert-registration-identifiant** (identificateur de consignation d'avertissement) (obligatoire): indique l'identificateur de consignation de l'action automatique `Auto-alert` (avertissement automatique) consignée qui a donné lieu à l'avertissement (voir 6.5.2 et 13.1);
- new-entry** (nouvelle entrée) (optionnel): achemine les informations provenant de la nouvelle entrée qui ont été demandées dans le paramètre de consignation d'avertissement automatique (voir 13.1). Ce paramètre peut être absent si l'utilisateur MS n'a pas spécifié la composante `requested-attributes` (attributs demandés) dans le paramètre de consignation d'avertissement automatique.

NOTE – Même si la composante `requested-attributes` (attributs demandés) n'est pas spécifiée dans le paramètre de consignation d'avertissement automatique, le paramètre `new-entry` (nouvelle entrée), s'il est présent, indiquera le numéro de séquence de l'entrée.

8.2.6.2 Résultat d'avertissement

Si la demande aboutit, le résultat d'avertissement **alert-result** sera renvoyé.

```

AlertResult ::= NULL

```

8.2.6.3 Erreurs abstraites d'avertissement

Si la demande échoue, l'erreur de sécurité sera signalée. Cela se produit lorsque l'opération abstraite transgresse la politique de sécurité en vigueur.

8.2.7 Opération abstraite Modify (modification)

L'opération abstraite **Modify** (modification) permet à l'utilisateur MS d'ajouter ou de retirer des attributs, ou des valeurs d'attribut individuelles associées à une ou plusieurs entrées de la mémoire de messages. Seuls certains types d'attributs généraux peuvent être modifiés au moyen de cette opération abstraite (voir 11.6). Les types d'attributs propres à un type de contenu particulier, et pouvant être modifiés, sont définis dans la Spécification appropriée. La définition d'un type d'attribut pouvant être modifié peut spécifier des restrictions concernant l'exécution de l'opération abstraite Modify. L'utilisation de l'opération abstraite Modify n'est pas normalisée avec les contextes d'application 1988.

Lorsque l'opération abstraite Modify est invoquée, la mémoire de messages vérifie d'abord que l'argument fourni est valable; si n'importe quelle erreur est trouvée, l'une des erreurs abstraites énumérées est renvoyée. Toutes les erreurs statiques, c'est-à-dire celles qui peuvent être détectées en examinant l'argument de modification, seront détectées avant de tenter d'appliquer les modifications spécifiées aux entrées sélectionnées; les erreurs statiques sont résumées à la fin du 8.2.7.1.

Si l'argument est valable, les modifications spécifiées sont appliquées à chacune des entrées sélectionnées dans l'ordre spécifié dans l'argument des entrées. Toutes les modifications seront appliquées à une entrée sélectionnée avant d'appliquer toute modification à l'entrée sélectionnée qui suit. Si la modification d'une entrée ne peut pas être exécutée, cette entrée est rétablie à son état d'origine et l'opération abstraite se termine par une erreur de modification. Cependant, les entrées dont la modification a abouti restent dans leur état modifié et sont signalées dans l'erreur de modification.

```

modify ABSTRACT-OPERATION ::= {
  ARGUMENT      ModifyArgument
  RESULT        ModifyResult
  ERRORS        {attribute-error | invalid-parameters-error | security-error |
                 sequence-number-error | service-error | modify-error | message-group-error |
                 entry-class-error | ms-extension-error,
                 ... -- réservé pour des extensions futures -- }
  CODE          op-modify }

```

8.2.7.1 Argument de modification

```

ModifyArgument ::= SET {
  entry-class   [0] EntryClass DEFAULT delivery,
  entries       CHOICE {
    selector     [1] Selector,
    specific-entries [2] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber },
  modifications [3] SEQUENCE SIZE (1..ub-modifications) OF EntryModification,
  modify-extensions [4] MSExtensions OPTIONAL }

```

Les paramètres de l'argument de modification **modify-argument** ont la signification suivante:

- entry-class** (classe d'entrée) (optionnel): ce paramètre spécifie la classe d'entrées qui contient les entrées à modifier. Si une classe d'entrées ne pouvant pas être modifiée est spécifiée, une erreur de classe d'entrées est créée;
- entries** (entrées) (obligatoire): les entrées à modifier sont identifiées soit par des sélecteurs **selector** (voir 8.1.3), soit explicitement par des numéros de séquence. Si la composante **specific-entries** (entrées propres) est présente, l'existence de chaque entrée sera vérifiée avant d'appliquer toute modification et une erreur de numéro de séquence sera signalée si une ou plusieurs entrées n'existent pas. Les entrées seront traitées dans l'ordre spécifié;
- modifications** (obligatoire): ce paramètre définit une séquence de modifications, qui seront appliquées à chaque entrée sélectionnée dans l'ordre spécifié. Les types d'attribut présents dans le paramètre seront examinés avant d'appliquer toute modification, et une erreur d'attribut sera signalée si un type d'attribut non pris en charge, ou ne pouvant être modifié, est spécifié.

```

EntryModification ::= SET {
  strict        [0] BOOLEAN DEFAULT FALSE,
  modification CHOICE {
    add-attribute [1] Attribute,
    remove-attribute [2] ATTRIBUTE.&id ({AttributeTable}),
    add-values     [3] OrderedAttribute,
    remove-values [4] OrderedAttribute } }

```

```

OrderedAttribute ::= SEQUENCE {
  attribute-type      ATTRIBUTE.&id ({AttributeTable}),
  attribute-values    SEQUENCE SIZE (1..ub-attribute-values) OF SEQUENCE {
    -- au moins une valeur doit être spécifiée --
    value             [0] ATTRIBUTE.&Type ({AttributeTable} {@attribute-type}) OPTIONAL,
    position          [1] INTEGER (1..ub-attribute-values) OPTIONAL } }

```

Les composantes du paramètre **entry-modification** (modification d'entrée) ont la signification suivante:

- 1) **strict** (optionnel): cette composante spécifie si une interprétation **stricte** des demandes de modification doit être suivie. Pour chaque type de modification défini ci-dessous, une ou plusieurs conditions d'erreurs mineures sont définies. Si la composante **strict** est à *vraie*, l'apparition de n'importe laquelle de ces erreurs mineures mettra fin à l'opération avec une erreur de modification. Si la composante **strict** est à *fausse*, l'exécution de l'opération abstraite continuera indépendamment de l'apparition d'une erreur mineure;
- 2) **modification** (obligatoire): cette composante indique la modification demandée.

S'il est spécifié **add-attribute** (ajout d'attribut), cette composante identifie un nouvel attribut à ajouter aux entrées sélectionnées, qui est entièrement spécifié par l'argument. Si une interprétation stricte est demandée, une tentative d'ajout d'un attribut déjà existant à une entrée met fin à l'opération abstraite avec une erreur de modification. Si l'interprétation stricte n'est pas demandée, le nouvel attribut remplace tout attribut existant.

S'il est spécifié **remove-attribute** (retrait d'attribut), cette composante identifie (par son type) un attribut à retirer des entrées sélectionnées. Si l'interprétation stricte est demandée, une tentative de retrait d'un attribut qui n'est pas présent dans une entrée entraîne la création d'une erreur de modification; sinon l'opération abstraite continue.

S'il est spécifié **add-values** (ajout de valeurs), cette composante identifie un attribut par le type d'attribut dans l'argument et une ou plusieurs valeurs à lui ajouter pour chaque entrée sélectionnée. Si l'interprétation stricte est demandée et si un attribut du type spécifié n'est pas déjà présent dans l'entrée, une erreur de modification est créée; sinon l'attribut est créé. Si l'interprétation stricte est demandée, une tentative d'ajout d'une valeur qui est déjà présente dans l'attribut entraîne la création d'une erreur de modification; sinon la valeur est ajoutée. Si l'interprétation stricte est demandée, une tentative d'ajout d'une seconde instance d'une valeur à un attribut qui n'autorise pas d'instances multiples de la même valeur entraîne la création d'une erreur de modification; sinon il n'est pas tenu compte de la demande. S'il est spécifié **position**, cette composante identifie la position à l'intérieur de la séquence de valeurs d'attribut que la valeur fournie occupera. Si une **position** est spécifiée qui dépasse le nombre de valeurs d'attribut présentes de plus d'une, une erreur de modification est créée, indépendamment de la valeur du paramètre strict. Si une valeur d'attribut occupe déjà la **position** spécifiée, cette valeur et toutes les valeurs qui la suivent sont déplacées. Lorsque le **position** est omise, les nouvelles valeurs sont ajoutées à la fin de la séquence des valeurs d'attribut. La première valeur d'un attribut occupe la position 1.

S'il est spécifié **remove-values** (retrait de valeurs), cette composante identifie un attribut par le type d'attribut dans l'argument et une ou plusieurs valeurs à retirer de cet attribut pour chaque entrée sélectionnée. Si l'interprétation stricte est demandée et si l'attribut ne figure pas dans l'entrée (ou s'il manque des valeurs spécifiées), une erreur de modification est créée; sinon l'opération abstraite continue. Si plus d'une valeur d'attribut est conforme à une valeur spécifiée, et qu'aucune composante **position** n'est présente, toutes ces valeurs d'attribut sont retirées. Si une composante **position** est présente, seule la valeur d'attribut à cette position est retirée; il n'est pas nécessaire de fournir la composante **value** (valeur) et il n'en est pas tenu compte si elle est présente. Si une **position** est spécifiée qui dépasse le nombre de valeurs d'attribut présentes, une erreur de modification est créée indépendamment de la valeur de la composante strict. Si toutes les valeurs d'attribut sont retirées, l'attribut est retiré de l'entrée.

Une erreur d'attribut est créée s'il est spécifié **add-values** (ajout de valeurs) ou **remove-values** (retrait de valeurs) pour un type d'attribut mono-valué. Cette erreur statique est détectée avant l'application de toute modification;

- d) **modify-extensions** (extensions de modification) (optionnel): ce paramètre permet de futures extensions générales et propres au contenu concernant l'argument de modification. Aucune extension n'est définie dans la présente Définition de service.

NOTE – Pour résumer, les erreurs statiques suivantes sont détectées avant toute tentative de modification:

- a) une entrée est spécifiée qui n'existe pas, ou qui appartient à une classe d'entrées ne pouvant pas être modifiée;
- b) une modification est demandée pour un type d'attribut non disponible (c'est-à-dire un type d'attribut non pris en charge par la mémoire de messages ou auquel l'utilisateur MS ne s'est pas abonné), ou pour un type d'attribut ne pouvant pas être modifié;
- c) une modification add-values (ajout de valeurs) ou remove-values (retrait de valeurs) est spécifiée pour un type d'attribut mono-valué;
- d) une demande est émise pour affecter un nom de groupe de messages non consigné à une entrée.

8.2.7.2 Résultat de modification

Si la demande aboutit, le résultat de modification **modify-result** sera renvoyé.

```
ModifyResult ::= SET {
    entries-modified          [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
    modify-result-extensions [1] MSExtensions OPTIONAL }
```

Les paramètres du résultat de modification **modify-result** ont la signification suivante:

- a) **entries-modified** (entrées modifiées) (conditionnel): numéros de séquence des entrées sélectionnées pour être modifiées. Présent si au moins une entrée a été sélectionnée pour être modifiée;

NOTE – Il est possible qu'une entrée sélectionnée pour être modifiée reste inchangée après application des modifications.
- b) **modify-result-extensions** (extensions du résultat de modification) (conditionnel): ce paramètre permet de futures extensions générales et propres au contenu concernant le résultat de modification. Aucune extension n'est définie dans la présente Définition de service.

8.2.7.3 Erreurs abstraites de modification

Si la demande échoue, l'une des erreurs abstraites énumérées sera signalée. Les conditions dans lesquelles telle ou telle erreur abstraite sera signalée sont définies à l'article 9.

8.3 Opérations abstraites de l'accès de dépôt MS

Les opérations abstraites suivantes sont disponibles à l'accès de dépôt MS:

- a) MS-message-submission (dépôt de message MS);
- b) MS-probe-submission (dépôt de message d'essai MS);
- c) MS-cancel-deferred-delivery (annulation de remise différée MS);
- d) MS-submission-control (commande de dépôt MS).

Ces opérations abstraites correspondent directement à celles fournies à l'accès de dépôt du service abstrait MTS, mais offrent une fonctionnalité supplémentaire liée aux capacités d'enregistrement de la mémoire de messages. Lorsqu'une opération abstraite MS-message-submission (dépôt de message MS) ou MS-probe-submission (dépôt de message d'essai MS) est invoquée, la mémoire de messages extrait les paramètres de l'argument de l'opération abstraite qui concernent l'activité propre à la mémoire de messages et, lorsqu'un dépôt est demandé, elle invoque l'opération correspondante à l'accès de dépôt MTS. Lorsque le dépôt auprès du système MTS n'est pas demandé, l'opération abstraite MS-message-submission a pour effet de créer une entrée dans la classe d'entrées Draft (brouillon).

8.3.1 Opération abstraite MS-message-submission (dépôt de message MS)

L'opération abstraite **MS-message-submission** (dépôt de message MS) sert à déposer un message auprès du système MTS et facultativement d'en enregistrer une copie, soit dans la classe d'entrées Submission-log (journal de dépôt), soit à la fois dans les classes d'entrée Submission (dépôt) et Submission-log. Le message peut également être enregistré (sans dépôt) dans la classe d'entrées Draft (brouillon). L'opération abstraite MS-message-submission utilise l'opération abstraite Message-submission (dépôt de message) du système MTS, qui est définie au 8.2.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

ms-message-submission ABSTRACT-OPERATION ::= {
  ARGUMENT      MSMessageSubmissionArgument
  RESULT        MSMessageSubmissionResult
  ERRORS        {submission-control-violated | element-of-service-not-subscribed |
                 originator-invalid | recipient-improperly-specified | inconsistent-request |
                 security-error | unsupported-critical-function | remote-bind-error,
                 ... -- extensions de 1994 --,
                 ms-extension-error | message-group-error | entry-class-error | service-error}
  CODE          op-ms-message-submission }

```

8.3.1.1 Argument du dépôt de message MS

```

MSMessageSubmissionArgument ::= SEQUENCE {
  COMPONENTS OF MessageSubmissionArgument -- ce type importé est étiqueté implicitement --,
  -- extension de 1994 --
  submission-options [4] MSSubmissionOptions OPTIONAL }

```

Les paramètres de l'argument de dépôt de message MS **MS-message-submission-argument** ont la signification suivante:

- a) **message-submission-argument** (argument de dépôt de message) (obligatoire): ce paramètre contient l'argument de l'opération abstraite Message-submission (dépôt de message) comme cela est défini au 8.2.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

La présente Définition de service définit une extension (voir 9.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4) à l'argument de dépôt de message du service abstrait MTS. Cette extension, **forwarding-request** (demande de retransmission), ne peut être présente que si un contexte d'application 1988 est utilisé:

```

forwarding-request EXTENSION ::= {
  SequenceNumber,
  IDENTIFIED BY standard-extension:36 }

```

Cette extension est présente si l'utilisateur MS demande la retransmission à d'autres utilisateurs d'un message remis enregistré dans la mémoire de messages. Elle indique le numéro de séquence de l'entrée à retransmettre. La demande est traitée comme cela est défini dans la Spécification concernant le type de contenu du message de retransmission. Une erreur intervenant dans le traitement de la demande de retransmission **forwarding-request** est signalée en utilisant l'erreur abstraite de demande incohérente du 8.2.2.7 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;

NOTE 1 – Lorsqu'un contexte d'application 1988 est utilisé, l'extension forwarding-request (demande de retransmission) fournit un mécanisme permettant de retransmettre les entrées delivered-message (message remis). Dans les contextes d'application 1994, un mécanisme propre au contenu peut être disponible pour permettre aux messages déposés d'insérer le contenu (en totalité ou en partie) des entrées de la classe Stored-message (message enregistré). On trouvera au 19.5.1 de la Rec. UIT-T X.420 | ISO/CEI 10021-7 une instance d'utilisation de chacun de ces mécanismes.

- b) **submission-options** (options de dépôt) (optionnel): ce paramètre contient les demandes de dépôt qui sont propres au fonctionnement de la mémoire de messages (voir 8.1.6). Si ces demandes indiquent que le dépôt auprès du système MTS est demandé, la mémoire de messages retire ce paramètre de l'argument avant d'invoquer l'opération abstraite Message-submission (dépôt de message). Si le paramètre est absent et si la présente association abstraite a identifié une consignation d'agent utilisateur dans son argument de rattachement MS, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt d'agent utilisateur relatives à cette consignation d'agent utilisateur; voir 8.2.5.1 g). Si le paramètre est absent et si la présente association abstraite n'a pas identifié de consignation d'agent utilisateur, ou si la consignation d'agent utilisateur ne contenait pas de valeur par défaut de dépôt d'agent utilisateur, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) consignées par l'opération abstraite Register-MS (consignation MS).

NOTE 2 – Lorsque les options de dépôt demandent la création d'une entrée draft-message (brouillon de message), l'argument de dépôt de message doit être complet syntaxiquement, afin d'être conforme à la syntaxe abstraite de l'opération abstraite.

8.3.1.2 Résultat du dépôt de message MS

Si la demande aboutit, le résultat de dépôt de message MS **MS-message-submission-result** sera renvoyé.

```

MSMessageSubmissionResult ::= CHOICE {
  mts-result SET {
    COMPONENTS OF      MessageSubmissionResult -- ce type importé est étiqueté implicitement --,
                      -- extension de 1994 --
    ms-message-result   [4] CommonSubmissionResults OPTIONAL },
  store-draft-result   [4] CommonSubmissionResults }

```

Les paramètres du résultat de dépôt de message MS **MS-message-submission-result** ont la signification suivante:

- a) **MTS-result** (résultat MTS) (conditionnel): ce paramètre n'est renvoyé que si l'utilisateur MS a demandé le dépôt du message auprès du système MTS et que ce dépôt a abouti. Il contient les composantes suivantes:
 - 1) **message-submission-result** (résultat du dépôt de message) (obligatoire): cette composante contient le résultat de l'opération abstraite Message-submission (dépôt de message) définie au 8.2.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
 - 2) **MS-message-result** (résultat du dépôt de message MS) (conditionnel): cette composante contient les résultats propres au fonctionnement de la mémoire de messages. Les composantes des résultats communs de dépôt seront présentes dans les cas définis au 8.1.7;
- b) **store-draft-result** (résultat de l'enregistrement du brouillon de message) (conditionnel): ce paramètre est renvoyé si l'utilisateur MS a demandé la création d'une entrée draft-message (brouillon de message) dans la classe d'entrées Draft (brouillon) et non le dépôt auprès du système MTS. La composante created-entry (entrée créée) indique le numéro de séquence de l'entrée draft-message nouvellement créée. Les autres composantes du paramètre store-draft-result sont définies au 8.1.7.

8.3.1.3 Erreurs abstraites concernant le dépôt de message MS

Si la demande échoue, l'une des erreurs abstraites définies à l'article 9 de la présente Définition de service, ou au 8.2.1.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4 sera signalée.

8.3.2 Opération abstraite MS-probe-submission (dépôt de message d'essai MS)

L'opération abstraite **MS-probe-submission** (dépôt de message d'essai MS) sert à déposer un message d'essai auprès du système MTS et facultativement d'en enregistrer une copie, soit dans la classe d'entrées Submission-log (journal de dépôt), soit à la fois dans les classes d'entrée Submission (dépôt) et Submission-log. L'opération abstraite MS-probe-submission (dépôt de message d'essai MS) utilise l'opération abstraite Probe-submission (dépôt de message d'essai) du système MTS, qui est définie au 8.2.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

ms-probe-submission ABSTRACT-OPERATION ::= {
  ARGUMENT  MSProbeSubmissionArgument
  RESULT    MSProbeSubmissionResult
  ERRORS    {submission-control-violated | element-of-service-not-subscribed |
             originator-invalid | recipient-improperly-specified | inconsistent-request |
             security-error | unsupported-critical-function | remote-bind-error,
             ... -- extensions de 1994 --,
             ms-extension-error | message-group-error | entry-class-error | service-error}
  CODE      op-ms-probe-submission }

```

8.3.2.1 Argument de dépôt de message d'essai MS

```

MSProbeSubmissionArgument ::= SET {
  COMPONENTS OF ProbeSubmissionArgument -- ce type importé est étiqueté implicitement --,
              -- extension de 1994 --
  submission-options [4] MSubmissionOptions OPTIONAL }

```

Les paramètres de l'argument de dépôt de message d'essai MS **MS-probe-submission-argument** ont la signification suivante:

- a) **probe-submission-argument** (argument de dépôt de message d'essai) (obligatoire): ce paramètre spécifie l'argument de l'opération abstraite Probe-submission (dépôt de message d'essai) comme cela est défini au 8.2.1.2.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;

- b) **submission-options** (options de dépôt) (optionnel): ce paramètre contient les demandes de dépôt qui sont propres au fonctionnement de la mémoire de messages (voir 8.1.6). La mémoire de messages retire ce paramètre de l'argument avant d'invoquer l'opération abstraite Probe-submission (dépôt de message d'essai). La composante object-entry-class (classe d'entrées de l'objet) de ce paramètre ne contiendra pas la valeur *draft* (brouillon); la spécification appropriée peut indiquer la mesure (le cas échéant) prise concernant la composante MS-submission-extensions (extensions du dépôt MS), si elle est présente. Si le paramètre submission-options est absent et si la présente association abstraite a identifié une consignation d'agent utilisateur dans son argument de rattachement MS, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt d'agent utilisateur relatives à cette consignation d'agent utilisateur; voir 8.2.5.1 g). Si le paramètre est absent et si la présente association abstraite n'a pas identifié de consignation d'agent utilisateur, ou si la consignation d'agent utilisateur ne contenait pas de valeur par défaut de dépôt d'agent utilisateur, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) consignées par l'opération abstraite Register-MS (consignation MS).

8.3.2.2 Résultat de dépôt de message d'essai MS

Si la demande aboutit, le résultat de dépôt de message d'essai MS **MS-probe-submission-result** est renvoyé.

```
MSProbeSubmissionResult ::= SET {
  COMPONENTS OF ProbeSubmissionResult -- ce type importé est étiqueté implicitement --,
  -- extension de 1994 --
  ms-probe-result [4] CommonSubmissionResults OPTIONAL }
```

Les paramètres du résultat de dépôt de message d'essai MS **MS-probe-submission-result** ont la signification suivante:

- a) **probe-submission-result** (résultat de dépôt de message d'essai) (obligatoire): ce paramètre est le résultat de l'opération abstraite Probe-submission (dépôt de message d'essai) comme cela est défini au 8.2.1.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) **ms-probe-result** (résultat du dépôt de message d'essai MS) (conditionnel): ce paramètre contient les résultats propres au fonctionnement de la mémoire de messages. Les composantes des résultats communs de dépôt seront présentes dans les cas définis au 8.1.7.

8.3.2.3 Erreurs abstraites concernant le dépôt de message d'essai MS

Si la demande échoue, l'une des erreurs abstraites définies à l'article 9 de la présente Définition de service, ou au 8.2.1.2.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4 sera signalée.

8.3.3 Opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS)

L'argument, le résultat et les erreurs de l'opération abstraite **MS-cancel-deferred-delivery** (annulation de remise différée MS) sont identiques à ceux de l'opération abstraite Cancel-deferred-delivery définie au 8.2.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
ms-cancel-deferred-delivery ABSTRACT-OPERATION ::= cancel-deferred-delivery
```

Si la demande d'une annulation de remise différée aboutit, la mémoire de messages cherche une entrée dans les classes d'entrées Submission (dépôt) et Submission-log (journal de messagerie) correspondant au message déposé pour lequel la remise différée a été annulée. Si cette entrée est présente, la mémoire de messages lui attachera un attribut deferred-delivery-cancellation-time (heure d'annulation de la remise différée) pour enregistrer la date et l'heure auxquelles l'annulation de la remise s'est produite, et mettra à jour l'attribut AC-report-summary (récapitulation de rapports AC) pour enregistrer l'annulation.

8.3.4 Opération abstraite MS-submission-control (commande de dépôt MS)

L'argument, le résultat et les erreurs de l'opération abstraite **MS-submission-control** (commande de dépôt MS) sont identiques à ceux de l'opération abstraite Submission-control définie au 8.2.1.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
ms-submission-control ABSTRACT-OPERATION ::= submission-control
```

9 Erreurs abstraites

Cet article définit les **erreurs abstraites** suivantes, associées à l'utilisation des opérations abstraites de l'accès d'extraction et de l'accès de dépôt MS:

- a) erreur d'attribut;
- b) erreur de demande d'action automatique;
- c) erreur de suppression;
- d) erreur de restriction de recherche;
- e) erreur de paramètres non valides;
- f) erreur d'intervalle;
- g) erreur de sécurité;
- h) erreur de numéro de séquence;
- i) erreur de service;
- j) erreur de groupe de messages;
- k) erreur d'extension MS;
- l) anciens pouvoirs mal spécifiés;
- m) nouveaux pouvoirs inacceptables;
- n) erreur de consignation MS;
- o) erreur de modification;
- p) erreur de classe d'entrée.

9.1 Priorité des erreurs

Lorsqu'une erreur est détectée au cours de l'exécution d'une opération abstraite, la mémoire de messages interrompra l'exécution de l'opération abstraite et renverra l'erreur à l'utilisateur MS. Lorsque plus d'une erreur se manifeste, les erreurs réelles renvoyées sont laissées au choix de la mémoire de messages.

NOTE – Lors de l'utilisation d'un contexte d'application 1988, il n'est pas indispensable que la mémoire de messages interrompe l'exécution de l'opération abstraite lorsqu'une erreur est rencontrée. Par conséquent, plusieurs types d'erreurs abstraites autorisent la signalisation de plus d'une erreur à la fois.

9.2 Erreur d'attribut

Une erreur d'attribut **Attribute-error** indique un problème lié à des attributs.

```

attribute-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems [0] SET SIZE (1..ub-per-entry) OF SET {
      problem [0] AttributeProblem,
      type [1] ATTRIBUTE.&id ({AttributeTable}),
      value [2] ATTRIBUTE.&Type ({AttributeTable} {@.type}) OPTIONAL } }
  CODE err-attribute-error }

AttributeProblem ::= INTEGER {
  invalid-attribute-value (0),
  unavailable-attribute-type (1),
  inappropriate-matching (2),
  attribute-type-not-subscribed (3),
  inappropriate-for-operation (4),
  -- extensions de 1994 --
  inappropriate-modification (5),
  single-valued-attribute (6) } (0..ub-error-reasons)

```

Ce paramètre a la signification suivante:

problems (problèmes) (obligatoire): problèmes particuliers rencontrés. Avec chaque problème, on indique le type d'attribut et, si nécessaire, pour éviter toute ambiguïté, la valeur qui est à l'origine du problème:

- a) *invalid-attribute-value* (type d'attribut non valide): une valeur visée d'attribut, spécifiée comme argument de l'opération abstraite, n'est pas conforme au type de données défini pour le type d'attribut en question;
- b) *unavailable-attribute-type* (type d'attribut indisponible): un type visé d'attribut, utilisé comme argument de l'opération abstraite, ne fait pas partie de ceux qui sont pris en charge par la mémoire de messages;
- c) *inappropriate-matching* (criblage inapproprié): le filtre contient un élément de filtre dans lequel un attribut est comparé à l'aide d'une règle de criblage [equality (égalité), ordering (ordonnancement), substrings (sous-chaînes) ou other-match (autre criblage)] qui n'est pas définie pour ce type d'attribut ou qui n'est pas prise en charge par la mémoire de messages;
- d) *attribute-type-not-subscribed* (type d'attribut auquel l'utilisateur ne s'est pas abonné): un type d'attribut utilisé comme argument de l'opération abstraite ne fait pas partie de ceux auxquels l'utilisateur MS s'est abonné;

NOTE – Une modification de l'abonnement n'est pas nécessairement reflétée dans les attributs présents dans une entrée créée avant la modification.

- e) *inappropriate-for-operation* (inapproprié pour l'utilisation): un type d'attribut utilisé comme argument de l'opération abstraite ne convient pas pour l'utilisation requise;
- f) *inappropriate-modification* (modification inappropriée): un type d'attribut ne pouvant pas être modifié a été utilisé comme argument de l'opération abstraite Modify (modification);
- g) *single-valued-attribute* (attribut mono-valué): on a tenté d'utiliser l'opération abstraite Modify (modification) pour appliquer les modifications add-values (ajout de valeurs) ou remove-values (retrait de valeurs) à un type d'attribut mono-valué.

9.3 Erreur de demande d'action automatique

Une erreur de demande d'action automatique **Auto-action-request-error** indique un problème lié à la consignation d'une action automatique.

```

auto-action-request-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems      [0] SET SIZE (1..ub-auto-registrations) OF SET {
      problem     [0] AutoActionRequestProblem,
      type        [1] AUTO-ACTION.&id ({AutoActionTable}) } }
  CODE          err-auto-action-request-error }

```

```

AutoActionRequestProblem ::= INTEGER {
  unavailable-auto-action-type      (0),
  auto-action-type-not-subscribed  (1),
  -- extension de 1994 --
  not-willing-to-perform            (2) } (0..ub-error-reasons)

```

Ce paramètre a la signification suivante:

problems (problèmes) (obligatoire): problèmes particuliers rencontrés. Avec chaque problème, on indique le type d'action automatique à l'origine du problème:

- a) *unavailable-auto-action-type* (type d'action automatique indisponible): un type d'action automatique utilisé comme argument de l'opération abstraite ne fait pas partie de ceux qui sont pris en charge par le fournisseur du service abstrait MS;
- b) *action-type-not-subscribed* (type d'action auquel l'utilisateur MS ne s'est pas abonné): un type d'action automatique utilisé comme argument de l'opération abstraite ne fait pas partie de ceux auxquels l'utilisateur MS s'est abonné;
- c) *not-willing-to-perform* (pas en mesure d'exécuter): la consignation d'action automatique est refusée parce que, si l'action était exécutée, cela conduirait à une utilisation excessive de ressources.

9.4 Erreur de suppression

Une erreur de suppression **Delete-error** indique un problème lié à une tentative pour supprimer une ou plusieurs entrées d'une classe d'entrée.

```
delete-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems      [0] SET SIZE (1..ub-messages) OF SET {
      problem      [0] DeleteProblem,
      sequence-number [1] SequenceNumber},
      -- extension de 1994 --
    entries-deleted [1] SET SIZE (1..ub-messages) OF SequenceNumber OPTIONAL }
  CODE      err-delete-error }

DeleteProblem ::= INTEGER {
  child-entry-specified      (0),
  delete-restriction-problem (1),
  -- extensions de 1994 --
  new-entry-specified        (2),
  entry-class-restriction    (3),
  stored-message-exists      (4) } (0..ub-error-reasons)
```

Le paramètre a la signification suivante:

- a) **problems** (problèmes) (obligatoire): problèmes particuliers rencontrés. Avec chaque problème, on indique le numéro de séquence de l'entrée à l'origine du problème:
 - 1) *child-entry-specified* (entrée fille spécifiée): on a tenté de supprimer une entrée fille;
 - 2) *delete-restriction-problem* (problème de restriction de suppression): on a tenté de transgresser une restriction spécifiée pour l'opération abstraite Delete (suppression) (voir 8.2.4);
 - 3) *new-entry-specified* (nouvelle entrée spécifiée): on a tenté de supprimer une entrée de la classe Delivery (remise) dont l'état d'extraction est *nouveau*;
 - 4) *entry-class-restriction* (restriction de classe d'entrée): on a tenté de supprimer une entrée d'une classe d'entrées pour laquelle la suppression est restreinte ou interdite;
 - 5) *stored-message-exists* (message enregistré existant): on a tenté de supprimer une entrée de la classe Message-log (journal de messagerie) pour laquelle une entrée correspondante existe dans la classe d'entrées Stored-message (message enregistré);
- b) **entries-deleted** (entrées supprimées) (conditionnel): ce paramètre identifie les entrées supprimées avant la fin de l'opération abstraite. Le paramètre est absent si aucune entrée n'est supprimée.

9.5 Erreur de restriction de recherche

Une erreur de restriction de recherche **Fetch-restriction-error** indique une tentative pour transgresser une restriction liée à l'opération abstraite Fetch (recherche).

```
fetch-restriction-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems [0] SET SIZE (1..ub-default-registrations) OF SET {
      problem [3] FetchRestrictionProblem,
      restriction CHOICE {
        content-type [0] OBJECT IDENTIFIER,
        eit [1] MS-EITs,
        attribute-length [2] INTEGER } } }
  CODE      err-fetch-restriction-error }

FetchRestrictionProblem ::= INTEGER {
  content-type-problem (1),
  eit-problem (2),
  maximum-length-problem (3) } (0..ub-error-reasons)
```

Ce paramètre a la signification suivante:

problems (problèmes) (obligatoire): problèmes particuliers rencontrés. Avec chaque problème, on indique le type de contenu, le type d'information codée ou la longueur de l'attribut qui est à l'origine du problème:

- a) *content-type-problem* (problème de type de contenu): le type de contenu du message recherché est interdit par les restrictions de recherche en vigueur;
- b) *EIT-problem* (problème de type d'information codée): les types d'information codée demandés dans l'opération abstraite Fetch (recherche) sont interdits par les restrictions de recherche en vigueur;
- c) *maximum-length-problem* (problème de longueur maximale): la longueur du codage d'une valeur d'attribut recherchée est supérieure à celle qui est autorisée par les restrictions de recherche en vigueur.

9.6 Erreur de paramètres non valides

Une erreur de paramètres non valides **Invalid-parameters-error** indique un problème indéfini intervenant dans le traitement de l'argument d'une opération abstraite. Cette erreur peut être utilisée, par exemple, pour indiquer qu'un paramètre facultatif a été utilisé dans un mauvais contexte ou qu'une valeur de l'un des paramètres est inappropriée.

```
invalid-parameters-error ABSTRACT-ERROR ::= {
    PARAMETER    NULL
    CODE         err-invalid-parameters-error }
```

Cette erreur a un paramètre de type néant.

9.7 Erreur d'intervalle

Une erreur d'intervalle **Range-error** indique un problème lié à l'intervalle spécifié dans un sélecteur comme argument d'une opération abstraite.

```
range-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problem    [0] RangeProblem}
    CODE         err-range-error }
```

```
RangeProblem ::= INTEGER {
    reversed    (0) } (0..ub-error-reasons)
```

Ce paramètre a la signification suivante:

problem (problème) (obligatoire): problème particulier rencontré. La valeur *reversed* (sens inverse) est renvoyée si la borne supérieure a indiqué un numéro de séquence inférieur ou une heure de création antérieure à ce qui a été indiqué par la borne inférieure.

Cette erreur abstraite ne sera pas créée lors de l'utilisation d'un contexte d'application 1994.

9.8 Erreur de sécurité

Une erreur de sécurité **Security-error** indique que l'opération abstraite demandée ne peut être assurée car elle transgresserait la politique de sécurité en vigueur. Cette erreur est définie au 8.2.2.8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

9.9 Erreur de numéro de séquence

Une erreur de numéro de séquence **Sequence-number-error** indique un problème lié au numéro de séquence spécifié dans l'argument d'une opération abstraite.

```

sequence-number-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems [1] SET SIZE (1..ub-messages) OF SET {
      problem [0] SequenceNumberProblem,
      sequence-number [1] SequenceNumber } }
  CODE err-sequence-number-error }

SequenceNumberProblem ::= INTEGER {
  no-such-entry (0) } (0..ub-error-reasons)

```

Ce paramètre a la signification suivante:

problems (problèmes) (obligatoire): problèmes particuliers rencontrés. Avec chaque problème, on indique le numéro de séquence à l'origine du problème. Si *no-such-entry* (nulle entrée semblable) est renvoyé, le numéro de séquence fourni ne correspond à aucune entrée de la classe spécifiée.

9.10 Erreur de service

Une erreur de service **Service-error** indique une erreur liée à la fourniture du service.

```

service-error ABSTRACT-ERROR ::= {
  PARAMETER ServiceErrorParameter
  CODE err-service-error }

ServiceErrorParameter ::= SET {
  problem [0] ServiceProblem,
  -- extension de 1994 --
  supplementary-information [1] GeneralString (SIZE (1..ub-supplementary-info-length)) OPTIONAL }

ServiceProblem ::= INTEGER {
  busy (0),
  unavailable (1),
  unwilling-to-perform (2) } (0..ub-error-reasons)

```

Ce paramètre a la signification suivante:

- a) **problem** (problème) (obligatoire): problème particulier rencontré:
 - 1) *busy* (occupé): la mémoire de messages, en totalité ou en partie, est trop occupée pour exécuter l'opération abstraite demandée mais pourra le faire plus tard;
 - 2) *unavailable* (indisponible): la mémoire de messages, en totalité ou en partie, est à l'heure actuelle non disponible;
 - 3) *unwilling-to-perform* (non prêt à exécuter): la mémoire de messages n'est pas prête à donner suite à cette demande car cela engendrerait une utilisation excessive de ressources;
- b) **supplementary-information** (information complémentaire) (conditionnel): fournit d'autres détails sur le problème signalé. Sa présence est laissée au choix du fournisseur du service MS.

9.11 Erreur de groupe de messages

Une erreur de groupe de messages **Message-group-error** indique un problème dans l'utilisation d'un nom de groupe de messages.

```

message-group-error ABSTRACT-ERROR ::= {
  PARAMETER MessageGroupErrorParameter
  CODE err-message-group-error }

MessageGroupErrorParameter ::= SET {
  problem [0] MessageGroupProblem,
  name [1] MessageGroupName }

```

```

MessageGroupProblem ::= INTEGER {
  name-not-registered      (0),
  name-already-registered (1),
  parent-not-registered   (2),
  group-not-empty        (3),
  name-in-use            (4),
  child-group-registered  (5),
  group-depth-exceeded    (6) } (0..ub-error-reasons)

```

Ce paramètre a la signification suivante:

- a) **problem** (problème) (obligatoire): identifie le problème rencontré. Les problèmes suivants peuvent être indiqués:
- 1) *name-not-registered* (nom non consigné): on a tenté d'affecter un nom de groupe de messages non consigné à l'attribut message-group-name (nom de groupe de messages) d'une entrée à l'aide de l'opération abstraite Modify (modification), ou de déconsigner un nom de groupe de messages non consigné à l'aide de l'opération abstraite Register-MS (consignation MS);
 - 2) *name-already-registered* (nom déjà consigné): on a tenté de consigner un nom de groupe de messages qui est déjà consigné;
 - 3) *parent-not-registered* (nom père non consigné): on a tenté de consigner un nom de groupe de messages contenant plusieurs parties de nom de groupe, dont le nom père n'est pas consigné (où le nom père est défini comme le nom formé en omettant la dernière partie du nom de groupe de messages);
 - 4) *group-not-empty* (groupe non vide): on a tenté de déconsigner un nom de groupe de messages qui figure dans au moins un attribut message-group-name (nom de groupe de messages) d'une entrée;
 - 5) *name-in-use* (nom déjà utilisé): on a tenté de déconsigner un nom de groupe de messages qui est référencé dans une action automatique Auto-modify (modification automatique) consignée;
 - 6) *child-group-registered* (nom de groupe fils consigné): on a tenté de déconsigner un nom de groupe de messages pour lequel au moins un nom de groupe de messages fils est actuellement consigné (où le nom fils est défini comme un nom formé en ajoutant une partie de nom de groupe à un nom de groupe de messages);
 - 7) *group-depth-exceeded* (profondeur de groupe dépassée): on a tenté de consigner un nom de groupe de messages contenant un plus grand nombre de parties de nom de groupe que le nombre maximal pris en charge par la mémoire de messages;
- b) **name** (nom) (obligatoire): indique le nom de groupe de messages qui fait l'objet de l'erreur abstraite.

9.12 Erreur d'extension MS

Une erreur d'extension MS **MS-extension-error** indique un problème concernant un paramètre d'extension présenté dans l'argument d'une opération abstraite.

```

ms-extension-error ABSTRACT-ERROR ::= {
  PARAMETER MSExtensionErrorParameter
  CODE      err-ms-extension-error }

MSExtensionErrorParameter ::= CHOICE {
  ms-extension-problem    [0] MSExtensionItem,
  unknown-ms-extension   [1] OBJECT IDENTIFIER }

```

Ce paramètre a la signification suivante:

- a) **ms-extension-problem** (problème d'extension MS) (conditionnel): identifie le problème rencontré. La définition d'une instance d'un problème **MS-extension-problem** peut apparaître dans la spécification pour un type de contenu donné;

NOTE – Les problèmes d'extension MS spécifiés pour la messagerie de personne à personne sont énumérés dans l'ensemble d'objets informationnels IP-submission-errors (erreurs de dépôt de message de personne à personne); voir 19.5.2.3 de la Rec. UIT-T X.420 | ISO/CEI 10021-7.

- b) **unknown-MS-extension** (extension MS inconnue) (conditionnel): une extension présentée dans l'argument d'une opération abstraite est inconnue de la mémoire de messages. Le paramètre contient la valeur de l'identificateur d'objet de l'extension inconnue obtenue à partir de son champ d'identification.

9.13 Erreur de consignation MS

Une erreur de consignation MS **Register-MS-error** indique un problème dans une tentative pour consigner des informations avec la mémoire de messages.

```
register-ms-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problem          [0] RegistrationProblem,
    registration-type [1] RegistrationTypes }
  CODE err-register-ms-error }
```

```
RegistrationProblem ::= ENUMERATED {
  registration-not-supported (0),
  registration-improperly-specified (1),
  registration-limit-exceeded (2),
  ... -- réservé pour des extensions futures -- }
```

Le paramètre a la signification suivante:

- a) **problem** (problème) (obligatoire): indique le problème particulier rencontré. Soit la consignation demandée ne fait pas partie de celles prises en charge par la mémoire de messages, ou la demande a été mal spécifiée, ou une limite pragmatique a été atteinte dans le nombre de consignations que la mémoire de messages est capable d'enregistrer;
- b) **registration-type** (type de consignation) (obligatoire): indique le type de consignation demandé par l'utilisateur MS auquel le problème est associé; voir 8.2.5.1 j) pour la définition des types de consignation.

NOTE – Lorsqu'une erreur abstraite différente fournit une description plus précise d'un problème de consignation, l'erreur Register-MS-error n'est pas signalée.

9.14 Anciens pouvoirs mal spécifiés

Une erreur abstraite **Old-credentials-incorrectly-specified** (anciens pouvoirs mal spécifiés) indique que les pouvoirs ne peuvent pas être modifiés parce que les pouvoirs actuels (anciens) ont été mal spécifiés. Cette erreur est définie au 8.4.2.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

9.15 Nouveaux pouvoirs inacceptables

Une erreur abstraite **New-credentials-unacceptable** (nouveaux pouvoirs inacceptables) indique que les pouvoirs ne peuvent pas être modifiés parce que les nouveaux pouvoirs sont inacceptables. Cette erreur est définie au 8.4.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

9.16 Erreur de modification

Une erreur de modification **Modify-error** indique un problème dans une tentative pour modifier les attributs d'une entrée.

```
modify-error ABSTRACT-ERROR ::= {
  PARAMETER ModifyErrorParameter
  CODE err-modify-error }
```

```
ModifyErrorParameter ::= SET {
  entries-modified [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
  failing-entry [1] SequenceNumber,
  modification-number [2] INTEGER,
  problem [3] ModifyProblem }
```

Le paramètre a la signification suivante:

- a) **entries-modified** (entrées modifiées) (conditionnel): numéros de séquence des entrées auxquelles les modifications appliquées ont abouti. Présent s'il existe au moins une entrée dont la modification a abouti;
- b) **failing-entry** (entrée dont la modification a échoué) (obligatoire): numéro de séquence de l'entrée pour laquelle une modification a échoué;
- c) **modification-number** (numéro de modification) (obligatoire): numéro identifiant la modification qui a entraîné la création de l'erreur, dans la séquence présentée dans le paramètre modifications de l'argument de modification. La première modification porte le numéro '1';
- d) **problem** (problème) (obligatoire): identifie le problème rencontré:

```
ModifyProblem ::= INTEGER {
    attribute-not-present      (0),
    value-not-present         (1),
    attribute-or-value-already-exists (2),
    invalid-position          (3),
    modify-restriction-problem (4) } (0..ub-error-reasons)
```

- 1) *attribute-not-present* (attribut absent): l'entrée ne possède pas l'un des types d'attribut spécifiés comme un argument de l'opération;
- 2) *value-not-present* (valeur absente): l'entrée ne possède pas l'une des valeurs d'attribut spécifiées comme un argument de l'opération;
- 3) *attribute-or-value-already-exists* (attribut ou valeur déjà existants): on a tenté d'ajouter un attribut qui existait déjà dans l'entrée, ou une valeur qui existait déjà dans l'attribut;
- 4) *invalid-position* (position non valable): l'attribut contient trop peu de valeurs pour que la modification soit valable;
- 5) *modify-restriction-problem* (problème de restriction de modification): on a tenté de transgresser une restriction spécifiée pour l'opération abstraite Modify (modification).

S'il n'existe pas d'entrée dont la modification a abouti et si une erreur autre que celles définies dans **problem** (problème) intervient, l'erreur abstraite appropriée sera renvoyée. Cependant, s'il existe des entrées dont la modification a abouti, la mémoire de messages renverra une erreur de modification qui spécifie les entrées modifiées et qui signale l'erreur.

9.17 Erreur de classe d'entrée

Une erreur de classe d'entrées **Entry-class-error** indique un problème dans une tentative pour appeler une classe d'entrées dans une opération abstraite.

```
entry-class-error ABSTRACT-ERROR ::= {
    PARAMETER EntryClassErrorParameter
    CODE      err-entry-class-error }
```

```
EntryClassErrorParameter ::= SET {
    entry-class [0] EntryClass,
    problem    [1] BIT STRING {
        unsupported-entry-class (0),
        entry-class-not-subscribed (1),
        inappropriate-entry-class (2) } }
```

Le paramètre a la signification suivante:

- a) **entry-class** (classe d'entrée) (obligatoire): cette composante indique la classe d'entrées spécifiée dans l'opération abstraite;
- b) **problem** (problème) (obligatoire): soit la classe d'entrées spécifiée ne fait pas partie de celles qui sont prises en charge par la mémoire de messages, soit elle ne fait pas partie de celles auxquelles l'utilisateur MS s'est abonné, soit elle est inappropriée pour l'opération abstraite demandée.

SECTION 3 – TYPES D'ATTRIBUTS GÉNÉRAUX, RÈGLES DE CRIBLAGE GÉNÉRALES ET TYPES D'ACTIONS AUTOMATIQUES GÉNÉRALES

10 Présentation générale

Le modèle informationnel de la mémoire de messages et les notions d'attribut, de règle de criblage et d'action automatique sont présentés aux 6.3 et 6.5. L'article 11 définit les **types d'attributs généraux** spécifiés pour la mémoire de messages. L'article 12 définit les **règles de criblage générales** spécifiées pour la mémoire de messages. L'article 13 définit les **types d'actions automatiques générales** spécifiés pour la mémoire de messages.

11 Types d'attributs généraux

Les types d'attributs généraux sont valables pour tous les types de contenu des messages. D'autres types d'attributs, qui sont propres au contenu, sont définis dans leur spécification respective; à titre d'exemple, les types d'attributs propres au service de messagerie de personne à personne (IPMS) pour la mémoire de messages sont définis à l'article 19 de la Rec. UIT-T X.420 | ISO/CEI 10021-7.

11.1 Présentation générale des types d'attributs généraux

Les attributs généraux définis pour les entrées de chacune des classes d'entrée sont énumérés dans les Tableaux 2 et 3. Ils proviennent essentiellement des paramètres des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai), Message-delivery (remise de message) et Report-delivery (remise de rapport) du service abstrait du système de transfert de messages (MTS) qui est défini à l'article 8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Ces attributs sont désignés en conséquence. Certains attributs généraux sont créés par la mémoire de messages, certains d'entre eux sont également mis à jour par la mémoire de messages.

Le Tableau 2 définit les caractéristiques suivantes pour chaque type d'attribut général:

- pour les entrées des classes d'entrée Delivery (remise) et Delivery-log (journal de remise), s'il s'agit d'un type d'attribut toujours présent, présent de façon conditionnelle ou absent respectivement dans une entrée delivered-message (message remis), delivered-report (rapport remis) ou returned-content (contenu renvoyé);
- pour les entrées des classes d'entrée Submission (dépôt) et Submission-log (journal de dépôt), s'il s'agit d'un type d'attribut toujours présent, présent de façon conditionnelle ou absent dans une entrée submitted-message (message déposé) ou submitted-probe (message d'essai déposé);
- pour les entrées de la classe Draft (brouillon), s'il s'agit d'un type d'attribut toujours présent, présent de façon conditionnelle ou absent respectivement dans une entrée draft-message (brouillon de message);
- si la prise en charge par la mémoire de messages est obligatoire ou facultative pour les entrées des classes d'entrée Stored-message (message enregistré) et Message-log (journal de messagerie);
- si le type d'attribut est mono-valué ou multi-valué;
- si les valeurs de l'attribut peuvent être renvoyées dans une opération abstraite List (listage) [et uniquement pour les entrées de la classe Delivery (remise), dans une opération abstraite Alert (avertissement)];
- si le type d'attribut peut faire l'objet de demandes de récapitulation dans une opération abstraite Summarize (récapitulation).

Le Tableau 3 définit des informations équivalentes pour les entrées de la classe Auto-action-log (journal d'actions automatiques).

On trouvera dans les conventions du 5.2 une description plus détaillée des classifications des Tableaux 2 et 3.

L'utilisateur MS peut disposer d'un type d'attribut facultatif pris en charge par une mémoire de messages sous réserve d'abonnement. Chaque type d'attribut facultatif peut faire l'objet d'un abonnement individuel.

Les types d'attribut définis comme étant présents dans une entrée delivered-message (message remis) peuvent ne pas être toujours présents dans une entrée fille. Les règles régissant la présence de types d'attribut dans des entrées filles pourront faire l'objet de compléments dans la Spécification définissant le type de contenu de l'entrée fille.

Tableau 2 – Types d'attributs généraux communs de la mémoire de messages

Nom du type d'attribut		Présence dans:						Prise en charge par la mémoire de messages		Mono-/multi-valué	Disponible pour List (listage)	Disponible pour Summarize (récapitulation)
		entrée delivered-message	entrée delivered-report	entrée returned-content	entrée submitted-message*	entrée submitted-probe*	entrée draft-message*	Classe d'entrées Stored-message	Classe d'entrées Message-log*			
AC-correlated-report-list*	liste des rapports corrélés AC	-	-	-	C	C	-	O	O	M	Y	N
AC-report-summary*	récapitulation des rapports AC	-	-	-	C	C	-	O	O	M	Y	N
AC-unrelated-report-list*	liste des rapports non corrélés AC	-	-	-	C	C	-	O	O	M	Y	N
Child-sequence-numbers	numéros de séquence des entrées filles	C	C	C	C	-	C	M	M	M	Y	N
Content	contenu	P	-	P	P	-	P	M	-	S	N	N
Content-confidentiality-algorithm-identifier	identificateur d'algorithme de confidentialité du contenu	C	-	-	C	-	C	O	O	S	Y	N
Content-correlator	corrélateur de contenu	-	C	-	C	C	C	O	O	S	Y	N
Content-identifier	identificateur de contenu	C	C	-	C	C	C	O	O	S	Y	N
Content-integrity-check	vérification d'intégrité du contenu	C	-	-	-	-	-	O	O	S	Y	N
Content-length	longueur du contenu	P	-	P	P	C	P	O	O	S	Y	N
Content-returned	contenu renvoyé	-	P	-	-	-	-	O	O	S	Y	Y
Content-type	type de contenu	P	C	C	P	P	P	M	M	S	Y	Y
Conversion-with-loss-prohibited	conversion avec perte interdite	C	-	-	C	C	C	O	O	S	Y	N
Converted-EITs	types d'information codée convertis	C	-	-	-	-	-	O	O	M	Y	N
Creation-time	heure de création	P	P	P	P	P	P	M	M	S	Y	N
Deferred-delivery-cancellation-time*	heure d'annulation de la remise différée	-	-	-	C	-	-	O	O	S	Y	N
Deferred-delivery-time*	heure de remise différée	-	-	-	C	-	C	O	O	S	Y	N
Deletion-time*	heure de suppression	C	C	-	C	C	-	-	M	S	Y	N
Delivered-EITs	types d'information codée remis	P	-	-	-	-	-	O	O	M	Y	N
Delivery-flags	fanions de remise	P	-	-	-	-	-	O	O	S	Y	N
DL-expansion-history	chronologie du développement de la liste de distribution	C	C	-	-	-	-	O	O	M	Y	N
DL-expansion-prohibited*	interdiction du développement de la liste de distribution	-	-	-	C	C	C	O	O	S	Y	N
Entry-type	type d'entrée	P	P	P	P	P	P	M	M	S	Y	Y
Internal-trace-information*	information de trace interne	C	C	-	-	-	-	O	O	M	N	N
Latest-delivery-time*	heure limite de remise	-	-	-	C	-	C	O	O	S	Y	N
Marked-for-deletion*	marqué pour être supprimé	C	C	-	C	C	C	O	O	S	Y	Y
Message-delivery-envelope	enveloppe de remise de message	P	-	-	-	-	-	M	-	S	N	N
Message-delivery-time	heure de remise de message	P	-	-	-	-	-	O	O	S	Y	N
Message-group-name*	nom du groupe de messages	C	C	-	C	C	C	O	O	M	Y	Y
Message-identifier	identificateur de message	P	-	-	P	P	-	O	O	S	Y	N
Message-notes*	notes de message	C	C	C	C	C	C	O	O	M	Y	N
Message-origin-authentication-check	vérification d'authentification de l'origine du message	C	-	-	C	-	C	O	O	S	Y	N
Message-security-label	étiquette de sécurité du message	C	C	-	C	C	C	O	O	S	Y	N
Message-submission-envelope*	enveloppe de dépôt de message	-	-	-	P	-	P	M	-	S	N	N
Message-submission-time	heure de dépôt du message	P	-	-	P	P	-	O	O	S	Y	N
Message-token	jeton de message	C	-	-	-	-	-	O	O	S	Y	N

Tableau 2 (fin) – Types d'attributs généraux communs de la mémoire de messages

Nom du type d'attribut		Présence dans:						Prise en charge par la mémoire de messages		Mono-/multi-valué	Disponible pour List (listage)	Disponible pour Summarize (récapitulation)
		entrée delivered-message	entrée delivered-report	entrée returned-content	entrée submitted-message*	entrée submitted-probe*	entrée draft-message*	Classe d'entrées Stored-message	Classe d'entrées Message-log*			
MS-originated*	origine MS	–	–	–	C	–	–	O	O	S	Y	Y
MS-submission-error*	erreur de dépôt MS	–	–	–	C	C	–	–	O	S	Y	N
Original-EITs	types d'information codée d'origine	C	C	–	C	C	C	O	O	M	Y	N
Originally-intended-recipient-name	nom du destinataire prévu au départ	C	–	–	–	–	–	O	O	S	Y	N
Originating-MTA-certificate*	certificat de l'agent MTA expéditeur	–	–	–	C	–	–	O	O	S	Y	N
Originator-certificate	certificat de l'expéditeur	C	–	–	C	C	–	O	O	S	Y	N
Originator-name	nom de l'expéditeur	P	–	–	P	P	P	O	O	S	Y	N
Originator-report-request*	demande de rapport par l'expéditeur	–	–	–	P	P	P	O	O	M	Y	N
Originator-return-address*	adresse de retour à l'expéditeur	C	–	–	C	–	C	O	O	S	Y	N
Other-recipient-names	noms d'autres destinataires	C	–	–	–	–	–	O	O	M	Y	N
Parent-sequence-number	numéro de séquence d'entrée mère	C	–	P	C	–	C	M	M	S	Y	N
Per-message-indicators*	indicateurs par message	–	–	–	P	P	P	O	O	S	Y	N
Per-recipient-message-submission-fields*	champs de dépôt de message par destinataire	–	–	–	P	–	P	O	O	M	N	N
Per-recipient-probe-submission-fields*	champs de dépôt de message d'essai par destinataire	–	–	–	–	P	–	O	O	M	N	N
Per-recipient-report-delivery-fields	champs de remise de rapport par destinataire	–	P	–	–	–	–	M	M	M	Y	N
Priority	priorité	P	–	–	P	–	P	O	O	S	Y	Y
Probe-origin-authentication-check*	vérification d'authentification de l'origine du message d'essai	–	–	–	–	C	–	O	O	S	Y	N
Probe-submission-envelope*	enveloppe de dépôt du message d'essai	–	–	–	–	P	–	O	–	S	N	N
Proof-of-delivery-request	demande de preuve de remise	C	–	–	–	–	–	O	O	S	Y	N
Proof-of-submission*	preuve de dépôt	–	–	–	C	–	–	O	O	S	Y	N
Recipient-names*	noms des destinataires	–	–	–	P	P	P	O	O	M	Y	N
Recipient-reassignment-prohibited*	interdiction de réaffectation de destinataire	–	–	–	C	C	C	O	O	S	Y	N
Redirection-history	chronologie du réacheminement	C	C	–	–	–	–	O	O	M	Y	N
Report-delivery-envelope	enveloppe de remise du rapport	–	P	–	–	–	–	M	–	S	N	N
Reporting-DL-name	nom de la liste de distribution rapporteur	–	C	–	–	–	–	O	O	S	Y	N
Reporting-MTA-certificate	certificat de l'agent MTA rapporteur	–	C	–	–	–	–	O	O	S	Y	N
Report-origin-authentication-check	vérification d'authentification de l'origine du rapport	–	C	–	–	–	–	O	O	S	Y	N
Retrieval-status	état d'extraction	P	P	P	P	P	P	M	M	S	Y	Y
Security-classification	classification de sécurité	C	C	–	C	C	C	O	O	S	Y	Y
Sequence-number	numéro de séquence	P	P	P	P	P	P	M	M	S	Y	N
Storage-period*	période d'enregistrement	C	C	–	C	C	C	O	O	S	Y	N
Storage-time*	heure d'enregistrement	C	C	–	C	C	C	O	O	S	Y	N
Subject-submission-identifiant	identificateur de dépôt du sujet	–	P	–	–	–	–	M	M	S	Y	N
This-recipient-name	nom de ce destinataire	P	–	–	–	–	–	O	O	S	Y	N
Trace-information*	information de trace	C	C	–	–	–	–	O	O	M	N	N

NOTE – Les types d'attribut et les colonnes du tableau marqués d'un astérisque (*) ne sont pas définis pour les contextes d'application 1988.

**Tableau 3 – Types d'attributs généraux pour la classe d'entrées Auto-action-log
(journal de messagerie)**

Nom du type d'attribut		Présence dans une entrée de la classe Auto-action-log	Prise en charge par MS	Mono-/ multi- valué	Disponible pour List (listage)	Disponible pour Summarize (récapitulation)
Auto-action-error	erreur d'action automatique	C	M	S	Y	N
Auto-action-registration-identifier	identificateur de consignation d'action automatique	P	M	S	Y	N
Auto-action-subject-entry	entrée de l'objet de l'action automatique	P	M	S	Y	N
Auto-action-type	type d'action automatique	P	M	S	Y	Y
Content-type	type de contenu	C	M	S	Y	Y
Creation-time	heure de création	P	M	S	Y	N
Entry-type	type d'entrée	P	M	S	Y	Y
Retrieval-status	état d'extraction	P	M	S	Y	Y
Sequence-number	numéro de séquence	P	M	S	Y	N
NOTE – La classe d'entrées Auto-action-log n'est pas définie pour les contextes d'application 1988.						

11.1.1 Prescriptions de prise en charge de types d'attributs généraux par la mémoire de messages

Les types d'attributs généraux dont la prise en charge par la mémoire de messages est obligatoire figurent dans les Tableaux 2 et 3. Lorsqu'un attribut est pris en charge, il sera pris en charge dans toutes les classes d'entrée pour lesquelles il est défini [à l'exception possible de la classe d'entrées Message-log (journal de messagerie)]. Tous les attributs pris en charge pourront être extraits par l'opération abstraite Fetch (recherche), et, lorsque c'est indiqué dans les Tableaux 2 et 3, par l'opération abstraite List (listage). Les types d'attributs qui peuvent être présents dans les entrées de la classe Delivery (remise) et qui sont disponibles pour l'opération abstraite List (listage) (voir Tableau 2) sont également disponibles pour l'opération abstraite Alert (avertissement).

La sélection d'une entrée [au moyen du paramètre Filter (filtre)] pour un type d'attribut pris en charge en utilisant l'une de ses règles de criblage sera disponible si la prise en charge est également demandée pour la règle de criblage en question.

NOTES

1 La prise en charge d'un type d'attribut ne nécessite pas en elle-même la prise en charge de toutes les règles de criblage définies pour ce type d'attribut; voir 12.5.

2 Pour certains types d'attributs définis au départ dans les versions de la présente Définition de service publiées avant 1994, la présente version définit des règles de criblage génériques supplémentaires. Dans les contextes d'application 1988, la prise en charge de ces cribles génériques supplémentaires n'est pas interdite. Cependant, certaines réalisations de mémoires de messages créeront une erreur d'attribut si on leur demande d'exécuter un criblage de ce type.

11.1.2 Prescriptions de prise en charge de types d'attributs généraux par l'utilisateur MS

Côté utilisateur MS, il n'existe aucune prescription relative à la prise en charge de n'importe quel type d'attribut général.

NOTE – L'utilisateur MS peut utiliser diverses stratégies pour extraire les informations qui seront visualisées par l'utilisateur. Par exemple, l'utilisateur MS peut choisir d'extraire l'enveloppe complète de remise du message, ou peut également extraire de façon sélective les attributs qui sont dérivés de l'enveloppe de remise du message. Etant donné que les deux approches sont valables, aucune méthode n'est prescrite.

11.2 Description des types d'attributs généraux

Les paragraphes suivants décrivent brièvement chaque type d'attribut général ainsi que sa syntaxe abstraite utilisant la syntaxe de la classe d'objets informationnels ATTRIBUTE (attribut) définie au 6.3.3.3.

NOTE – Certains attributs généraux sont utilisés essentiellement pour le listage et le filtrage, alors que d'autres peuvent contenir des informations plus complexes, voire plus importantes (types de données ASN.1 plus structurés). Seuls quelques attributs généraux conviennent pour être utilisés avec l'opération abstraite Summarize (récapitulation).

11.2.1 Liste des rapports corrélés AC (AC-correlated-report-list)

Cet attribut général, qui est multi-valué, identifie les entrées delivered-report (rapport remis) qui ont été corrélées avec les destinataires spécifiés au départ d'un message ou d'un message d'essai déposé. L'attribut contient une valeur pour chaque valeur de l'argument *recipient-name* (nom de destinataire) de l'opération abstraite Message-submission (dépôt de message) ou Probe-submission (dépôt de message d'essai). Les valeurs se trouvant à des positions correspondantes dans cet attribut et dans l'attribut recipient-names (noms des destinataires) font référence au même destinataire. Chaque valeur indique les numéros de séquence des entrées delivered-report (rapport remis) appropriées, et les positions dans leur attribut per-recipient-report-delivery-fields (champs de remise de rapport par destinataire) des valeurs qui sont liées à ce destinataire spécifié. L'attribut est créé par la mémoire de messages. L'abonnement à cet attribut nécessite l'abonnement à l'action automatique Auto-correlate-reports (corrélation automatique des rapports); voir 13.3.

```
ms-ac-correlated-report-list ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX  ReportLocation,
  NUMERATION             multi-valued,
  ID                     id-att-ac-correlated-report-list }

ReportLocation ::= CHOICE {
  no-correlated-reports [0] NULL,
  location               [1] SEQUENCE OF PerRecipientReport }

PerRecipientReport ::= SEQUENCE {
  report-entry [0] SequenceNumber,
  position     [1] INTEGER (1..ub-recipients) DEFAULT 1 }
```

Les composantes du paramètre **report-location** (position des rapports) ont la signification suivante:

- a) **no-correlated-reports** (aucun rapport corrélé) (conditionnel): aucun rapport n'a été reçu qui puisse être corrélé avec ce destinataire spécifié au départ du message ou du message d'essai déposé. Il s'agit de la valeur initiale de l'attribut;
- b) **location** (position) (conditionnel): cette composante identifie les numéros de séquence des rapports remis et les positions des valeurs dans leurs attributs per-recipient-report-delivery-fields (champs de remise de rapport par destinataire) qui sont liés à ce destinataire spécifié au départ.

L'attribut est créé lorsque le message ou le message d'essai est déposé et il est mis à jour quand les rapports sont remis. Le contenu de cet attribut n'est affecté par aucune suppression ultérieure des entrées delivered-report (rapport remis) auxquelles il fait référence.

11.2.2 Récapitulation des rapports AC (AC-report-summary)

Cet attribut général, qui est multi-valué, contient une récapitulation des rapports demandés et des rapports reçus de chaque destinataire spécifié au départ d'un message ou d'un message d'essai déposé. L'attribut contient une valeur pour chaque valeur de l'argument *recipient-name* (nom de destinataire) de l'opération abstraite Message-submission (dépôt de message) ou Probe-submission (dépôt de message d'essai). Les valeurs se trouvant à des positions correspondantes dans cet attribut et dans l'attribut recipient-names (noms des destinataires) font référence au même destinataire. Les valeurs initiales de l'attribut sont fixées selon les valeurs du paramètre originator-report-request (demande de rapport par l'expéditeur) spécifié pour chaque destinataire, et sont modifiées quand les rapports sont remis. L'attribut est créé par la mémoire de messages. L'abonnement à cet attribut nécessite l'abonnement à l'action automatique Auto-correlate-reports (corrélation automatique des rapports); voir 13.3.

```
ms-ac-report-summary ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX  ReportSummary,
  EQUALITY MATCHING-RULE integerMatch,
  ORDERING MATCHING-RULE integerOrderingMatch,
  NUMERATION             multi-valued,
  ID                     id-att-ac-report-summary }

ReportSummary ::= ENUMERATED {
  no-report-requested           (0) -- rapport de non-remise supprimé --,
  no-report-received           (1) -- rapport de non-remise demandé --,
  report-outstanding           (2) -- rapport de remise demandé --,
  delivery-cancelled           (3),
  delivery-report-from-another-recipient (4),
  non-delivery-report-from-another-recipient (5),
  delivery-report-from-intended-recipient (6),
  non-delivery-report-from-intended-recipient (7) }
```

Les valeurs du paramètre **report-summary** (récapitulation de rapport) ont la signification suivante:

- a) *no-report-requested* (aucun rapport demandé): la suppression des rapports de non-remise a été demandée pour ce destinataire spécifié au départ et aucun rapport n'a été reçu. Il s'agit d'une valeur initiale possible de l'attribut;
- b) *no-report-received* (aucun rapport reçu): le destinataire spécifié au départ a demandé de signaler les non-remises et aucun rapport n'a été reçu. Il s'agit d'une valeur initiale possible de l'attribut;
- c) *report-outstanding* (rapport en attente): le destinataire spécifié au départ a demandé de signaler les remises et les non-remises et aucun rapport n'a été reçu. Il s'agit d'une valeur initiale possible de l'attribut;
- d) *delivery-cancelled* (remise annulée): aucun rapport ne sera reçu parce que la remise du message a été annulée au moyen de l'opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS);
- e) *delivery-report-from-another-recipient* (rapport de remise d'un autre destinataire): un rapport de remise a été reçu concernant le destinataire spécifié au départ qui indique que le message a bien été remis (voire que le message sujet du message d'essai a bien été remis) à un autre destinataire (éventuellement après réacheminement ou développement de la liste de distribution);
- f) *non-delivery-report-from-another-recipient* (rapport de non-remise d'un autre destinataire): un rapport de non-remise a été reçu concernant le destinataire spécifié au départ qui indique que la remise du message (voire que la remise du message sujet du message d'essai) à un autre destinataire a échoué (éventuellement après réacheminement ou développement de la liste de distribution);
- g) *delivery-report-from-intended-recipient* (rapport de remise du destinataire prévu): un rapport a été remis qui confirme la remise au destinataire spécifié au départ;
- h) *non-delivery-report-from-intended-recipient* (rapport de non-remise du destinataire prévu): un rapport a été remis qui indique la non-remise au destinataire spécifié au départ.

L'attribut est créé lorsque le message ou le message d'essai est déposé. Quand des rapports sont remis à la mémoire de messages, la valeur de l'attribut général AC-report-summary (récapitulation des rapports AC) est mise à jour si la valeur résultante est supérieure à celle qui est déjà enregistrée. On considère que plusieurs rapports peuvent être créés comme résultat d'une remise du système de remise physique ou comme résultat d'un développement de la liste de distribution. On ne recommande aucune méthode par laquelle une mémoire de messages détermine le destinataire prévu auquel un rapport fait référence et la méthode dépend donc de la réalisation. Le contenu de cet attribut n'est affecté par aucune suppression ultérieure des entrées delivered-report (rapport remis) auxquelles il fait référence.

NOTE – Une mémoire de messages qui est située au même endroit qu'un agent de transfert de message (MTA) peut utiliser la composante originally-specified-recipient-number (nombre de destinataires spécifiés au départ) de l'attribut per-recipient-report-transfer-fields (champs de transfert de rapport par destinataire) pour déterminer le destinataire prévu qui est le sujet d'un rapport entrant.

11.2.3 Liste des rapports non corrélés AC (AC-uncorrelated-report-list)

Cet attribut général, qui est multi-valué, identifie chacun des champs de remise de rapport par destinataire qui ont été corrélés avec un message ou un message d'essai déposé mais qui n'ont été corrélés avec aucun de ses destinataires spécifiés au départ. Il est créé par la mémoire de messages. L'abonnement à cet attribut nécessite l'abonnement à l'action automatique Auto-correlate-reports (corrélation automatique des rapports); voir 13.3.

```
ms-ac-uncorrelated-report-list ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX      PerRecipientReport,
    NUMERATION                 multi-valued,
    ID                          id-att-ac-uncorrelated-report-list }
```

L'attribut est créé quand le message ou le message d'essai est déposé, et il est mis à jour quand des rapports du genre décrit sont remis. Le contenu de cet attribut n'est affecté par aucune suppression ultérieure des entrées delivered-report (rapport remis) auxquelles il fait référence.

11.2.4 Erreur d'action automatique (auto-action-error)

Cet attribut général identifie l'erreur d'action automatique qui s'est produite quand une action automatique consignée a été appliquée à une entrée. L'attribut ne peut être présent que dans les entrées de la classe Auto-action-log (journal d'actions automatiques); voir 6.5.3. Il est créé par la mémoire de messages.

```
ms-auto-action-error ATTRIBUTE ::= {
    WITH ATTRIBUTE-SYNTAX      AutoActionError,
    NUMERATION                 single-valued,
    ID                          id-att-auto-action-error }
```

11.2.5 Identificateur de consignation d'action automatique (auto-action-registration-identifiant)

Cet attribut général indique l'identificateur de consignation de l'action automatique dont cette entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques) est l'objet de l'exécution. L'action automatique consignée est identifiée par la combinaison de cet attribut et de la valeur correspondante de l'attribut général auto-action-type (type d'action automatique). L'attribut ne peut être présent que dans les entrées de la classe Auto-action-log (journal d'actions automatiques); voir 6.5.2. Il est créé par la mémoire de messages.

```
ms-auto-action-registration-identifiant ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      INTEGER,
  EQUALITY MATCHING-RULE    integerMatch,
  ORDERING MATCHING-RULE    integerOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-auto-action-registration-identifiant }
```

11.2.6 Entrée de l'objet de l'action automatique (auto-action-subject-entry)

Cet attribut général indique le numéro de séquence de l'entrée de la classe Stored-message (message enregistré) qui était l'objet de l'exécution de l'action automatique enregistrée par la présente entrée. L'attribut ne peut être présent que dans les entrées de la classe Auto-action-log (journal d'actions automatiques). Il est créé par la mémoire de messages.

```
ms-auto-action-subject-entry ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      SequenceNumber,
  EQUALITY MATCHING-RULE    integerMatch,
  ORDERING MATCHING-RULE    integerOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-auto-action-subject-entry }
```

11.2.7 Type d'action automatique (auto-action-type)

Cet attribut général identifie le type d'action automatique dont cette entrée est l'objet de l'exécution. L'action automatique consignée est identifiée par la combinaison de cet attribut et de la valeur correspondante de l'attribut général auto-action-registration-identifiant (identificateur de consignation de l'action automatique). L'attribut ne peut être présent que dans les entrées de la classe Auto-action-log (journal d'actions automatiques); voir 6.5.1. Il est créé par la mémoire de messages.

```
ms-auto-action-type ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      AutoActionType,
  EQUALITY MATCHING-RULE    objectIdentifierMatch,
  NUMERATION                 single-valued,
  ID                          id-att-auto-action-type }
```

11.2.8 Numéros de séquence des entrées filles (child-sequence-numbers)

Cet attribut général, qui est multi-valué, identifie les entrées filles d'une entrée mère, le cas échéant. Il est créé par la mémoire de messages. Cet attribut est présent dans une entrée à laquelle sont associées une ou plusieurs entrées filles. Il ne figure pas dans une entrée à laquelle n'est associée aucune entrée fille; voir 6.3.4.

```
ms-child-sequence-numbers ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      SequenceNumber,
  NUMERATION                 multi-valued,
  ID                          id-att-child-sequence-numbers }
```

11.2.9 Contenu (content)

Cet attribut général contient le contenu *content* complet d'un message tel qu'il est déposé par l'opération abstraite MS-message-submission (dépôt de message MS) ou tel qu'il est remis par l'opération abstraite Message-delivery (remise de message) ou tel qu'il est présenté dans le contenu renvoyé *returned-content* de l'opération abstraite Report-delivery (remise de rapport). Dans ce dernier cas, l'attribut général de contenu est créé dans l'entrée fille returned-content (contenu renvoyé) et non pas dans l'entrée delivered-report (rapport remis) proprement dite; voir 8.2.1.1.1.37 et 8.3.1.2.1.14 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Le codage du contenu *content* figurant dans la chaîne d'octets qui constitue cet attribut doit être identique, lorsqu'il est extrait, à celui remis à la mémoire de messages; il n'existe aucune contrainte pour le codage utilisé pour la chaîne d'octets contenant elle-même.

```
mt-content ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      Content,
  NUMERATION                 single-valued,
  ID                          id-att-content }
```

11.2.10 Identificateur d'algorithme de confidentialité du contenu (content-confidentiality-algorithm-identifiant)

Cet attribut général contient l'argument *content-confidentiality-algorithm-identifiant* (identificateur d'algorithme de confidentialité du contenu) des opérations abstraites Message-submission (dépôt de message) et Message-delivery (remise de message); voir 8.2.1.1.1.27 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-content-confidentiality-algorithm-identifiant ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      AlgorithmIdentifier,
  NUMERATION                 single-valued,
  ID                          id-att-content-confidentiality-algorithm-identifiant }
```

11.2.11 Corrélateur de contenu (content-correlator)

Cet attribut général contient l'argument *content-correlator* (corrélateur de contenu) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai) et Report-delivery (remise de rapport); voir 8.2.1.1.1.36 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-content-correlator ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ContentCorrelator,
  EQUALITY MATCHING-RULE    contentCorrelatorMatch,
  NUMERATION                 single-valued,
  ID                          id-att-content-correlator }
```

11.2.12 Identificateur de contenu (content-identifiant)

Cet attribut général contient l'argument *content-identifiant* (identificateur de contenu) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai), Message-delivery (remise de message) et Report-delivery (remise de rapport). Il peut être créé par l'expéditeur; voir 8.2.1.1.1.35 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-content-identifiant ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ContentIdentifier,
  EQUALITY MATCHING-RULE    contentIdentifierMatch,
  NUMERATION                 single-valued,
  ID                          id-att-content-identifiant }
```

11.2.13 Vérification d'intégrité du contenu (content-integrity-check)

Cet attribut général contient l'argument *content-integrity-check* (vérification de l'intégrité du contenu) de l'opération abstraite Message-delivery (remise de message): il fournit à l'utilisateur MS un moyen de vérifier que le contenu d'un message remis n'a pas été modifié. Il peut être créé par l'expéditeur du message. Dans les entrées submitted-message (message déposé), cette information est contenue dans l'attribut per-recipient-message-submission-fields (champs de dépôt de message par destinataire); voir 8.2.1.1.1.28 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-content-integrity-check ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ContentIntegrityCheck,
  NUMERATION                 single-valued,
  ID                          id-att-content-integrity-check }
```

11.2.14 Longueur du contenu (content-length)

Cet attribut général indique la longueur du contenu en octets, d'un message déposé, d'un message d'essai déposé, d'un brouillon de message, d'un message remis ou du contenu renvoyé d'un rapport remis. Pour un message d'essai déposé, l'attribut indique la longueur en octets du contenu du message sujet. Il est créé par la mémoire de messages.

```
ms-content-length ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ContentLength,
  ORDERING MATCHING-RULE    integerOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-content-length }
```

11.2.15 Contenu renvoyé (content-returned)

Cet attribut général indique si un contenu a été renvoyé dans l'opération abstraite Report-delivery (remise de rapport). Il est créé par la mémoire de messages; voir 6.3.4.

```
ms-content-returned ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      BOOLEAN,
  EQUALITY MATCHING-RULE    booleanMatch,
  NUMERATION                 single-valued,
  ID                          id-att-content-returned }
```

11.2.16 Type de contenu (content-type)

Cet attribut général, qui identifie le type de contenu d'un message, est créé à partir de l'argument *content-type* (type de contenu) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai), Message-delivery (remise de message) et Report-delivery (remise de rapport); voir 8.2.1.1.1.34 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Lorsque cet attribut est présent dans une entrée de la classe Auto-action-log (journal d'actions automatiques), il identifie le type de contenu de l'entrée qui était l'objet de l'exécution de l'action automatique; lorsque l'entrée est un rapport remis, l'attribut est présent de façon conditionnelle.

```
mt-content-type ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    OBJECT IDENTIFIER,
  EQUALITY MATCHING-RULE  objectIdentifierMatch,
  NUMERATION              single-valued,
  ID                      id-att-content-type }
```

11.2.17 Conversion avec perte interdite (conversion-with-loss-prohibited)

Cet attribut général contient l'argument *conversion-with-loss-prohibited* (conversion avec perte interdite) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai) et Message-delivery (remise de message). Il indique si la conversion avec perte d'information a été autorisée ou non par l'expéditeur; voir 8.2.1.1.1.10 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-conversion-with-loss-prohibited ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    ConversionWithLossProhibited,
  EQUALITY MATCHING-RULE  integerMatch,
  NUMERATION              single-valued,
  ID                      id-att-conversion-with-loss-prohibited }
```

11.2.18 Types d'information codée convertis (converted EITs)

Cet attribut général, qui est multi-valué, identifie les types d'information codée du contenu après conversion tels qu'ils sont indiqués par l'opération abstraite Message-delivery (remise de message). Il est créé à partir de l'argument *converted-encoded-information-types* (types d'information codée convertis) de l'opération abstraite Message-delivery (remise de message). Il est absent si aucune conversion n'a eu lieu; voir 8.3.1.1.1.8 et 8.3.1.2.1.5 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-converted-EITs ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MS-EIT,
  EQUALITY MATCHING-RULE  objectIdentifierMatch,
  NUMERATION              multi-valued,
  ID                      id-att-converted-EITs }
```

11.2.19 Heure de création (creation-time)

Cet attribut général indique l'heure à laquelle l'entrée a été créée dans la mémoire de messages. Il est créé par la mémoire de messages; voir 6.3.2.

NOTE – Deux entrées consécutives ou plus peuvent avoir la même heure de création.

```
ms-creation-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    CreationTime,
  EQUALITY MATCHING-RULE  uTCTimeMatch,
  ORDERING MATCHING-RULE  uTCTimeOrderingMatch,
  NUMERATION              single-valued,
  ID                      id-att-creation-time }
```

11.2.20 Heure d'annulation de la remise différée (deferred-delivery-cancellation-time)

Cet attribut général indique l'heure à laquelle le système MTS a confirmé que l'annulation de la remise différée d'un message préalablement déposé a abouti; voir 8.3.3. Il est créé par la mémoire de messages.

```
ms-deferred-delivery-cancellation-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    DeferredDeliveryCancellationTime,
  EQUALITY MATCHING-RULE  uTCTimeMatch,
  ORDERING MATCHING-RULE  uTCTimeOrderingMatch,
  NUMERATION              single-valued,
  ID                      id-att-deferred-delivery-cancellation-time }
```

DeferredDeliveryCancellationTime ::= UTCTime

11.2.21 Heure de remise différée (deferred-delivery-time)

Cet attribut général contient l'argument *deferred-delivery-time* (heure de remise différée) de l'opération abstraite Message-submission (dépôt de message). Il indique l'heure avant laquelle un message déposé ne doit pas être remis à son ou ses destinataires; voir 8.2.1.1.1.12 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. L'abonnement à cet attribut nécessite l'abonnement à l'action automatique Auto-correlate-reports (corrélation automatique des rapports); voir 13.3.

```
mt-deferred-delivery-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      DeferredDeliveryTime,
  EQUALITY MATCHING-RULE    uTCTimeMatch,
  ORDERING MATCHING-RULE    uTCTimeOrderingMatch,
  NUMERATION                 single-valued,
  ID                         id-att-deferred-delivery-time }
```

11.2.22 Heure de suppression (deletion-time)

Cet attribut général peut être présent dans les entrées de la classe Message-log (journal de messagerie), et indique l'heure à laquelle l'entrée correspondante dans la classe d'entrées Stored-message (message enregistré) a été supprimée. Dans le cas d'une entrée de la classe Submission-log (journal de dépôt) pour laquelle aucune entrée n'est créée dans la classe d'entrées Submission (dépôt), l'attribut deletion-time est créé lorsque l'entrée de la classe Submission-log est créée et on lui affecte la même valeur que celle affectée à l'heure de création de l'entrée. Il est créé par la mémoire de messages; voir 16.1.4.

```
ms-deletion-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      DeletionTime,
  EQUALITY MATCHING-RULE    uTCTimeMatch,
  ORDERING MATCHING-RULE    uTCTimeOrderingMatch,
  NUMERATION                 single-valued,
  ID                         id-att-deletion-time }
```

DeletionTime ::= UTCTime

11.2.23 Types d'information codée remis (delivered EITs)

Cet attribut général, qui est multi-valué, identifie les types d'information codée dans le contenu d'un message tels qu'ils sont remis. Il est créé par la mémoire de messages et est dérivé des arguments *original-encoded-information-types* (types d'information codée d'origine) et *converted-encoded-information-types* (types d'information codée convertis) de l'opération abstraite Message-delivery (remise de message).

```
ms-delivered-EITs ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MS-EIT,
  EQUALITY MATCHING-RULE    objectIdentifierMatch,
  NUMERATION                 multi-valued,
  ID                         id-att-delivered-EITs }
```

NOTE – Cet attribut a la même valeur que celle de l'argument *converted-encoded-information-types* (types d'information codée convertis) (s'il est présent); sinon, la même valeur que celle de l'argument *original-encoded-information-types* (types d'information codée d'origine). Il fournit une méthode pratique pour déterminer, en une seule opération, les types d'information codée présents dans le contenu.

11.2.24 Fanions de remise (delivery-flags)

Cet attribut général contient des informations dérivées des arguments de l'opération abstraite Message-delivery (remise de message). Il indique si la conversion implicite du contenu est interdite; voir 8.2.1.1.1.9 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-delivery-flags ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      DeliveryFlags,
  EQUALITY MATCHING-RULE    bitStringMatch,
  NUMERATION                 single-valued,
  ID                         id-att-delivery-flags }
```

11.2.25 Chronologie du développement de la liste de distribution (DL-expansion-history)

Cet attribut général, qui est multi-valué, contient une chronologie du développement de la liste de distribution. S'il est présent dans un message remis, il contient un ou plusieurs noms de liste de distribution utilisés pendant le processus de développement. Il est absent si la remise à l'utilisateur MS n'a nécessité aucun développement de la liste de distribution. S'il est présent dans un rapport remis, il contient le nom de l'expéditeur et un ou plusieurs noms de listes de distribution utilisés pendant le processus de développement. Il est absent si le message ou si le message sujet du rapport n'a pas nécessité de développement d'une liste de distribution; voir 8.3.1.1.1.7 et 8.3.1.2.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-dl-expansion-history ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      DLExpansion,
  OTHER MATCHING-RULES      {redirectionOrDLExpansionMatch |
                             redirectionOrDLExpansionElementsMatch |
                             redirectionOrDLExpansionSubstringElementsMatch},
  NUMERATION                 multi-valued,
  ID                          id-att-dl-expansion-history }
```

11.2.26 Interdiction du développement de la liste de distribution (DL-expansion-prohibited)

Cet attribut général contient l'argument *DL-expansion-prohibited* (interdiction du développement de la liste de distribution) des opérations abstraites Message-submission (dépôt de message) et Probe-submission (dépôt de message d'essai); il indique si le développement de la liste de distribution à l'intérieur du système MTS est interdit pour un nom de destinataire qui indique une liste de distribution; voir 8.2.1.1.1.6 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-dl-expansion-prohibited ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      DLExpansionProhibited,
  EQUALITY MATCHING-RULE     integerMatch,
  NUMERATION                 single-valued,
  ID                          id-att-dl-expansion-prohibited }
```

11.2.27 Type d'entrée (entry-type)

Cet attribut général identifie le type d'une entrée; voir 6.3.6. Il est créé par la mémoire de messages.

```
ms-entry-type ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      EntryType,
  EQUALITY MATCHING-RULE     integerMatch,
  ORDERING MATCHING-RULE     integerOrderingMatch, -- règle non définie dans les contextes
                                                                    d'application 1988 --
  NUMERATION                 single-valued,
  ID                          id-att-entry-type }
```

11.2.28 Information de trace interne (internal-trace-information)

Cet attribut général, qui est multi-valué, fournit des détails sur les mesures concernant un message (ou un message d'essai ou un rapport) prises par chaque agent MTA par lequel il transite pendant son transfert à l'intérieur d'un domaine de gestion. Il est créé à partir des attributs Message-delivery-envelope (enveloppe de remise de message) et Report-delivery-envelope (enveloppe de remise de rapport); voir 12.2.1.1.1.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-internal-trace-information ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      InternalTraceInformationElement,
  NUMERATION                 multi-valued,
  ID                          id-att-internal-trace-information }
```

11.2.29 Heure limite de remise (latest-delivery-time)

Cet attribut général contient l'argument *latest-delivery-time* (heure limite de remise) de l'opération abstraite Message-submission (dépôt de message); il indique l'heure après laquelle un message ne doit pas être remis à son ou ses destinataires; voir 8.2.1.1.1.13 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-latest-delivery-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      LatestDeliveryTime,
  EQUALITY MATCHING-RULE     uTCTimeMatch,
  ORDERING MATCHING-RULE     uTCTimeOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-latest-delivery-time }
```

11.2.30 Marqué pour être supprimé (marked-for-deletion)

Cet attribut général indique (par sa présence ou son absence) si l'utilisateur MS a marqué l'entrée pour la supprimer ultérieurement. La prise en charge de cet attribut suppose que l'utilisateur MS est en mesure de le créer ou de le modifier au moyen de l'opération abstraite Modify (modification) et de l'action automatique Auto-modify (modification automatique).

```
ms-marked-for-deletion ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    NULL,
  NUMERATION              single-valued,
  ID                      id-att-marked-for-deletion }
```

NOTE – L'utilisateur MS peut utiliser ce type d'attribut pour prendre en charge une fonction de suppression de l'entrée en deux étapes, c'est-à-dire une 'poubelle'.

11.2.31 Enveloppe de remise de message (message-delivery-envelope)

Cet attribut général contient la composante *message-delivery-envelope* (enveloppe de remise de message) de l'argument de l'opération abstraite Message-delivery (remise de message); voir la Figure 2 (partie 5) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-message-delivery-envelope ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MessageDeliveryEnvelope,
  NUMERATION              single-valued,
  ID                      id-att-message-delivery-envelope }
```

11.2.32 Heure de remise du message (message-delivery-time)

Cet attribut général contient l'argument *message-delivery-time* (heure de remise de message) de l'opération abstraite Message-delivery (remise de message); voir 8.3.1.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

NOTE – Il n'y a pas d'attribut général correspondant au paramètre delivery-time (heure de remise) de l'opération abstraite Report-delivery (remise de rapport) car, pour être utile, ce paramètre doit être corrélé avec le nom du destinataire auquel le message a été remis. Les deux informations sont présentes dans l'attribut général per-recipient-report-delivery-fields (champs de remise de rapports par destinataire).

```
mt-message-delivery-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MessageDeliveryTime,
  EQUALITY MATCHING-RULE  uTCTimeMatch,
  ORDERING MATCHING-RULE  uTCTimeOrderingMatch,
  NUMERATION              single-valued,
  ID                      id-att-message-delivery-time }
```

11.2.33 Nom de groupe de messages (message-group-name)

Cet attribut général, qui est multi-valué, identifie les noms de groupe de messages que l'utilisateur MS a attachés à une entrée; voir 6.4. La prise en charge de cet attribut général suppose que l'utilisateur MS est en mesure de le créer ou de le modifier en utilisant l'opération abstraite Modify (modification) et l'action automatique Auto-modify (modification automatique). Une tentative pour affecter une valeur de l'attribut message-group-name (nom de groupe de messages) qui n'a pas été préalablement consignée au moyen de l'opération abstraite Register-MS (consignation MS) se traduira par la création d'une erreur de groupe de messages. Toutes les valeurs de l'attribut doivent être distinctes.

```
ms-message-group-name ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MessageGroupName,
  EQUALITY MATCHING-RULE  mSStringListMatch,
  OTHER MATCHING-RULES    {mSSingleSubstringListMatch |
                           mSStringListElementsMatch |
                           mSSingleSubstringListElementsMatch |
                           valueCountMatch},
  NUMERATION              multi-valued,
  ID                      id-att-message-group-name }
```

NOTE – Le nombre maximal de parties de nom de groupe qui sera présent dans un nom de groupe de messages peut être déterminé à partir du paramètre message-group-depth (profondeur du groupe de messages) du résultat de rattachement MS; voir 7.1.2 h).

11.2.34 Identificateur de message (message-identifiant)

Cet attribut général contient un identificateur MTS qui permet de distinguer ce message ou ce message d'essai de tous les autres messages et messages d'essai déposés, ainsi que de tous les autres messages remis. Il contient le résultat *message-submission-identifiant* (identificateur de dépôt de message) ou *probe-submission-identifiant* (identificateur de dépôt de message d'essai) de l'opération abstraite Message-submission (dépôt de message) ou Probe-submission (dépôt de message d'essai), ou l'argument *message-delivery-identifiant* (identificateur de remise de message) de l'opération abstraite Message-delivery (remise de message). Il est créé par le système MTS; voir 8.2.1.1.2.1, 8.2.1.2.2.1, et 8.3.1.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-message-identifiant ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MTSIdentifiant,
  EQUALITY MATCHING-RULE    mTSIdentifiantMatch, -- règle non définie dans les contextes d'application
                                                                    1988 --
  NUMERATION                single-valued,
  ID                        id-att-message-identifiant }

```

11.2.35 Notes de message (message-notes)

Cet attribut général, qui est multi-valué, contient les annotations textuelles spécifiées par l'utilisateur MS. Le sens attaché aux notes du message ne concerne que l'utilisateur MS. La prise en charge de cet attribut suppose que l'utilisateur MS est en mesure de le créer ou de le modifier au moyen de l'opération abstraite Modify (modification) et de l'action automatique Auto-modify (modification automatique).

NOTE – L'attribut peut contenir des mots de passe, des annotations, ou des mementos choisis par l'utilisateur MS, mais ne représente pas des informations acheminées par le système MTS.

```

ms-message-notes ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      GeneralString (SIZE (1..ub-message-notes-length)),
  EQUALITY MATCHING-RULE    mSStringMatch,
  SUBSTRINGS MATCHING-RULE mSSubstringsMatch,
  NUMERATION                multi-valued,
  ID                        id-att-message-notes }

```

11.2.36 Vérification d'authentification de l'origine du message (message-origin-authentication-check)

Cet attribut général contient l'argument *message-origin-authentication-check* (vérification d'authentification de l'origine du message) des opérations abstraites Message-submission (dépôt de message) et Message-delivery (remise de message). Il permet au ou aux destinataires d'un message d'authentifier son origine et peut être créé par l'expéditeur du message; voir 8.2.1.1.1.29 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-message-origin-authentication-check ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MessageOriginAuthenticationCheck,
  NUMERATION                single-valued,
  ID                        id-att-message-origin-authentication-check }

```

11.2.37 Etiquette de sécurité du message (message-security-label)

Cet attribut général contient l'argument *message-security-label* (étiquette de sécurité du message) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai), Message-delivery (remise de message) et Report-delivery (remise de rapport). Il comprend un ensemble d'attributs de sécurité parmi lesquels un identificateur de politique de sécurité, une classification de sécurité, une marque de confidentialité et un ensemble de catégories de sécurité; voir 8.2.1.1.1.30 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-message-security-label ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MessageSecurityLabel,
  NUMERATION                single-valued,
  ID                        id-att-message-security-label }

```

11.2.38 Enveloppe de dépôt de message (message-submission-envelope)

Cet attribut général contient la composante *enveloppe* (enveloppe) de l'argument de l'opération abstraite Message-submission (dépôt de message); voir la Figure 2 (partie 3) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-message-submission-envelope ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MessageSubmissionEnvelope,
  NUMERATION                single-valued,
  ID                        id-att-message-submission-envelope }

```

11.2.39 Heure de dépôt du message (message-submission-time)

Cet attribut général indique l'heure à laquelle le système MTS a accepté de prendre la responsabilité du message ou du message d'essai. Il contient le résultat *message-submission-time* (heure de dépôt du message) de l'opération abstraite Message-submission (dépôt de message), ou le résultat *probe-submission-time* (heure de dépôt du message d'essai) de l'opération abstraite Probe-submission (dépôt de message d'essai), ou l'argument *message-submission-time* de l'opération abstraite Message-delivery (remise de message). Il est créé par le système MTS; voir 8.2.1.1.2.2 et 8.2.1.2.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-message-submission-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MessageSubmissionTime,
  EQUALITY MATCHING-RULE  uTCTimeMatch,
  ORDERING MATCHING-RULE uTCTimeOrderingMatch,
  NUMERATION              single-valued,
  ID                       id-att-message-submission-time }

```

11.2.40 Jeton de message (message-token)

Cet attribut général contient l'argument *message-token* (jeton de message) de l'opération abstraite Message-delivery (remise de message). Il est créé par l'expéditeur du message; voir 8.2.1.1.1.26 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Dans les entrées submitted-message (message déposé), cette information figure dans l'attribut per-recipient-message-submission-fields (champs de dépôt de message par destinataire).

```

mt-message-token ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    MessageToken,
  NUMERATION              single-valued,
  ID                       id-att-message-token }

```

11.2.41 Origine MS (MS-originated)

Cet attribut général indique si le message représenté par une entrée submitted-message (message déposé) a été déposé par la mémoire de messages à la suite de l'exécution d'une action automatique, ou s'il a été déposé par l'utilisateur MS. L'attribut est présent dans le premier cas et absent dans le second. Il est créé par la mémoire de messages.

```

ms-originated ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    NULL,
  NUMERATION              single-valued,
  ID                       id-att-ms-originated }

```

11.2.42 Erreur de dépôt MS (MS-submission-error)

Cet attribut général peut être présent dans les entrées de la classe Submission-log (journal de dépôt) et indique l'erreur renvoyée par l'opération abstraite MS-message-submission (dépôt de message MS) ou MS-probe-submission (dépôt de message d'essai MS) qui a entraîné la création de l'entrée. Il est créé par la mémoire de messages.

```

ms-submission-error ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX    SubmissionError,
  NUMERATION              single-valued,
  ID                       id-att-ms-submission-error }

SubmissionError ::= CHOICE {
  submission-control-violated    [1] NULL,
  originator-invalid            [2] NULL,
  recipient-improperly-specified [3] ImproperlySpecifiedRecipients,
  element-of-service-not-subscribed [4] NULL,
  inconsistent-request         [11] NULL,
  security-error                [12] SecurityProblem,
  unsupported-critical-function  [13] NULL,
  remote-bind-error            [15] NULL,
  service-error                 [27] ServiceErrorParameter,
  message-group-error          [30] MessageGroupErrorParameter,
  ms-extension-error           [31] MSExtensionErrorParameter,
  entry-class-error            [34] EntryClassErrorParameter }

```

Chaque composante du paramètre submission-error (erreur de dépôt) correspond à l'une des erreurs définies pour les opérations abstraites MS-message-submission et MS-probe-submission. Les numéros d'étiquettes correspondent aux codes d'erreur distante; voir 8.3.1 et 8.3.2.

11.2.43 Types d'information codée d'origine (original-EITs)

Cet attribut général, qui est multi-valué, identifie les types d'information codée figurant dans le contenu d'un message déposé, ou spécifiés dans l'enveloppe d'un message d'essai déposé. Il est créé à partir de l'argument *original-encoded-information-types* (types d'information codée d'origine) des opérations abstraites Message-delivery (remise de message), Report-delivery (remise de rapport), Message-submission (dépôt de message) et Probe-submission (dépôt de message d'essai); voir 8.2.1.1.1.33 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-original-EITs ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      MS-EIT,
  EQUALITY MATCHING-RULE    objectIdentifierMatch,
  NUMERATION                 multi-valued,
  ID                          id-att-original-EITs }
```

11.2.44 Nom de destinataire prévu au départ (originally-intended-recipient-name)

Cet attribut général contient l'argument *originally-intended-recipient-name* (nom de destinataire prévu au départ) de l'opération abstraite Message-delivery (remise de message), et il est présent si le message a été réacheminé; voir 8.3.1.1.1.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-originally-intended-recipient-name ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ORName,
  EQUALITY MATCHING-RULE    oRNameMatch,
  OTHER MATCHING-RULES     {oRNameElementsMatch | oRNameSubstringElementsMatch |
                             oRNameSingleElementMatch},
  NUMERATION                 single-valued,
  ID                          id-att-originally-intended-recipient-name }
```

11.2.45 Certificat de l'agent MTA expéditeur (originating-MTA-certificate)

Cet attribut général contient le résultat *originating-MTA-certificate* (certificat de l'agent MTA expéditeur) de l'opération abstraite Message-submission (dépôt de message). Il est créé par l'agent MTA auprès duquel le message a été déposé; voir 8.2.1.1.2.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-originating-MTA-certificate ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      OriginatingMTACertificate,
  NUMERATION                 single-valued,
  ID                          id-att-originating-MTA-certificate }
```

11.2.46 Certificat de l'expéditeur (originator-certificate)

Cet attribut général contient l'argument *originator-certificate* (certificat de l'expéditeur) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message) et Message-delivery (remise de message). Il est créé par une personne de confiance (par exemple une autorité de certification) et peut être fourni par l'expéditeur du message; voir 8.2.1.1.1.25 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-originator-certificate ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      OriginatorCertificate,
  NUMERATION                 single-valued,
  ID                          id-att-originator-certificate }
```

11.2.47 Nom de l'expéditeur (originator-name)

Cet attribut général contient l'argument *originator-name* (nom de l'expéditeur) des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message) et Message-delivery (remise de message); voir 8.2.1.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-originator-name ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ORName,
  EQUALITY MATCHING-RULE    oRNameMatch,
  OTHER MATCHING-RULES     {oRNameElementsMatch | oRNameSubstringElementsMatch |
                             oRNameSingleElementMatch},
  NUMERATION                 single-valued,
  ID                          id-att-originator-name }
```

11.2.48 Demande de rapport par l'expéditeur (originator-report-request)

Cet attribut général, qui est multi-valué, contient l'argument *originator-report-request* (demande de rapport par l'expéditeur) des opérations abstraites Message-submission (dépôt de message) et Probe-submission (dépôt de message d'essai); voir 8.2.1.1.1.22 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. L'ordonnancement des valeurs de cet attribut correspond à l'ordonnancement des valeurs de l'attribut per-recipient-message-submission-fields (champs de dépôt de message par destinataire).

```
mt-originator-report-request ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   OriginatorReportRequest,
  NUMERATION              multi-valued,
  ID                      id-att-originator-report-request }
```

11.2.49 Adresse de retour à l'expéditeur (originator-return-address)

Cet attribut général contient l'argument *originator-return-address* (adresse de retour à l'expéditeur) des opérations abstraites Message-submission (dépôt de message) et Message-delivery (remise de message); il indique l'adresse OR postale de l'expéditeur du message; voir 8.2.1.1.1.21 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-originator-return-address ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   OriginatorReturnAddress,
  NUMERATION              single-valued,
  ID                      id-att-originator-return-address }
```

11.2.50 Noms d'autres destinataires (other-recipient-names)

Cet attribut général, qui est multi-valué, contient l'argument *other-recipient-names* (noms d'autres destinataires) de l'opération abstraite Message-delivery (remise de message); voir 8.3.1.1.1.6 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-other-recipient-names ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   ORName,
  EQUALITY MATCHING-RULE oRNameMatch,
  OTHER MATCHING-RULES   {oRNameElementsMatch | oRNameSubstringElementsMatch |
                           oRNameSingleElementMatch},
  NUMERATION              multi-valued,
  ID                      id-att-other-recipient-names }
```

11.2.51 Numéro de séquence d'entrée mère (parent-sequence-number)

Cet attribut général identifie l'entrée mère de l'entrée. Il est créé par la mémoire de messages. Il est toujours présent dans une entrée fille et absent dans une entrée principale; voir 6.3.4.

```
ms-parent-sequence-number ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   SequenceNumber,
  EQUALITY MATCHING-RULE integerMatch,
  ORDERING MATCHING-RULE integerOrderingMatch,
  NUMERATION              single-valued,
  ID                      id-att-parent-sequence-number }
```

11.2.52 Indicateurs par message (per-message-indicators)

Cet attribut général comprend divers arguments de l'opération abstraite Message-submission (dépôt de message) et les deux premiers arguments de l'opération abstraite Probe-submission (dépôt de message d'essai). Ces arguments comprennent les arguments *implicit-conversion-prohibited* (conversion implicite interdite), *alternate-recipient-allowed* (autre destinataire autorisé), *disclosure-of-other-recipients* (divulcation des autres destinataires), et *content-return-request* (demande de renvoi du contenu); voir la Figure 2 (partie 10) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-per-message-indicators ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   PerMessageIndicators,
  EQUALITY MATCHING-RULE bitStringMatch,
  NUMERATION              single-valued,
  ID                      id-att-per-message-indicators }
```

11.2.53 Champs de dépôt de message par destinataire (per-recipient-message-submission-fields)

Cet attribut général, qui est multi-valué, contient la composante *per-recipient-fields* (champs par destinataire) de l'attribut Message-submission-enveloppe (enveloppe de dépôt du message); voir la Figure 2 (partie 7) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-per-recipient-message-submission-fields ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   PerRecipientMessageSubmissionFields,
  NUMERATION              multi-valued,
  ID                      id-att-per-recipient-message-submission-fields }
```

11.2.54 Champs de dépôt de message d'essai par destinataire (per-recipient-probe-submission-fields)

Cet attribut général, qui est multi-valué, contient la composante *per-recipient-fields* (champs par destinataire) de l'attribut Probe-submission-enveloppe (enveloppe de dépôt du message d'essai); voir la Figure 2 (partie 8) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-per-recipient-probe-submission-fields ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   PerRecipientProbeSubmissionFields,
  NUMERATION              multi-valued,
  ID                      id-att-per-recipient-probe-submission-fields }
```

11.2.55 Champs de dépôt de rapport par destinataire (per-recipient-report-delivery-fields)

Cet attribut général, qui est multi-valué, contient la composante *per-recipient-fields* (champs par destinataire) de l'attribut Report-submission-enveloppe (enveloppe de dépôt du rapport); voir la Figure 2 (partie 9) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-per-recipient-report-delivery-fields ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   PerRecipientReportDeliveryFields,
  NUMERATION              multi-valued,
  ID                      id-att-per-recipient-report-delivery-fields }
```

11.2.56 Priorité (priority)

Cet attribut général contient l'argument *priority* (priorité) des opérations abstraites Message-submission (dépôt de message) et Message-delivery (remise de message). Si aucune valeur pour le paramètre n'est fournie dans l'opération abstraite Message-submission (dépôt de message) ou Message-delivery (remise de message), l'attribut est créé avec la valeur par défaut de priorité; voir 8.2.1.1.1.8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-priority ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   Priority,
  EQUALITY MATCHING-RULE integerMatch,
  ORDERING MATCHING-RULE integerOrderingMatch, -- règle non définie dans les contextes
                                                    d'application 1988 --
  NUMERATION              single-valued,
  ID                      id-att-priority }
```

11.2.57 Vérification d'authentification de l'origine du message d'essai (probe-origin-authentication-check)

Cet attribut général contient l'argument *probe-origin-authentication-check* (vérification d'authentification de l'origine du message d'essai) de l'opération abstraite Probe-submission (dépôt de message d'essai). Il fournit à tout agent MTA par lequel le message d'essai transite un moyen d'en authentifier l'origine. Il peut être créé par l'utilisateur MS lorsqu'il expédie le message d'essai; voir 8.2.1.2.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-probe-origin-authentication-check ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   ProbeOriginAuthenticationCheck,
  NUMERATION              single-valued,
  ID                      id-att-probe-origin-authentication-check }
```

11.2.58 Enveloppe de dépôt de message d'essai (probe-submission-enveloppe)

Cet attribut général contient la composante *enveloppe* (enveloppe) de l'argument de l'opération abstraite Probe-submission (dépôt de message d'essai); voir la Figure 2 (partie 3) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-probe-submission-enveloppe ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX   ProbeSubmissionEnvelope,
  NUMERATION              single-valued,
  ID                      id-att-probe-submission-enveloppe }
```

11.2.59 Demande de preuve de remise (proof-of-delivery-request)

Cet attribut général contient l'argument *proof-of-delivery-request* (demande de preuve de remise) de l'opération abstraite Message-delivery (remise de message). Il peut être créé par l'expéditeur du message. Dans les entrées submitted-message (message déposé), cette information peut être trouvée dans l'attribut per-recipient-message-submission-fields (champs de dépôt du message par destinataire); voir 8.2.1.1.1.32 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-proof-of-delivery-request ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ProofOfDeliveryRequest,
  EQUALITY MATCHING-RULE    integerMatch, -- règle non définie dans les contextes d'application
                                     1988 --
  NUMERATION                single-valued,
  ID                        id-att-proof-of-delivery-request }

```

11.2.60 Preuve de dépôt (proof-of-submission)

Cet attribut général contient le résultat *proof-of-submission* (preuve de dépôt) de l'opération abstraite Message-submission (dépôt de message). Il fournit à l'utilisateur MS une preuve de dépôt d'un message auprès du système MTS; voir 8.2.1.1.2.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-proof-of-submission ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ProofOfSubmission,
  NUMERATION                single-valued,
  ID                        id-att-proof-of-submission }

```

11.2.61 Noms des destinataires (recipient-names)

Cet attribut général, qui est multi-valué, contient les composantes *recipient-name* (nom de destinataire) spécifiées dans les composantes *per-recipient-fields* (champs par destinataire) des attributs Message-submission-envelope (enveloppe de dépôt de message) et Probe-submission-envelope (enveloppe de dépôt de message d'essai). Il est créé par la mémoire de messages. L'ordonnancement des valeurs de cet attribut correspond à l'ordonnancement des valeurs de l'attribut per-recipient-message-submission-fields (champs de dépôt de message par destinataire) {ou per-recipient-probe-submission-fields (champs de dépôt de message d'essai par destinataire)}; voir 8.2.1.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

ms-recipient-names ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ORName,
  EQUALITY MATCHING-RULE    oRNameMatch,
  OTHER MATCHING-RULES     {oRNameElementsMatch | oRNameSubstringElementsMatch |
                                     oRNameSingleElementMatch},
  NUMERATION                multi-valued,
  ID                        id-att-recipient-names }

```

11.2.62 Interdiction de réaffectation de destinataire (recipient-reassignment-prohibited)

Cet attribut général contient l'argument *recipient-reassignment-prohibited* (interdiction de réaffectation de destinataire) des opérations abstraites Message-submission (dépôt de message) et Probe-submission (dépôt de message d'essai). Il indique si un message ou un message d'essai déposé par l'utilisateur MS peut être réacheminé vers un autre destinataire choisi par le destinataire, consigné par le destinataire prévu; voir 8.2.1.1.1.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-recipient-reassignment-prohibited ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      RecipientReassignmentProhibited,
  EQUALITY MATCHING-RULE    integerMatch,
  NUMERATION                single-valued,
  ID                        id-att-recipient-reassignment-prohibited }

```

11.2.63 Chronologie du réacheminement (redirection-history)

Cet attribut général, qui est multi-valué, contient la chronologie du ou des réacheminements vers des destinataires, ainsi que le ou les motifs de ces réacheminements provenant des opérations abstraites Message-delivery (remise de message) ou Report-delivery (remise de rapport). Dans le dernier cas, les réacheminements concernent le rapport de remise lui-même et non le message sujet du rapport; voir 8.3.1.1.1.5 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-redirect-history ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      Redirection,
  OTHER MATCHING-RULES      {redirectionOrDLExpansionMatch |
                             redirectionOrDLExpansionElementsMatch |
                             redirectionOrDLExpansionSubstringElementsMatch |
                             redirectionReasonMatch},
  NUMERATION                 multi-valued,
  ID                          id-att-redirect-history }

```

11.2.64 Enveloppe de remise de rapport (report-delivery-envelope)

Cet attribut général contient la composante *report-delivery-envelope* (enveloppe de remise de rapport) de l'argument de l'opération abstraite Report-delivery (remise de rapport); voir la Figure 2 (partie 9) de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-report-delivery-envelope ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ReportDeliveryEnvelope,
  NUMERATION                 single-valued,
  ID                          id-att-report-delivery-envelope }

```

11.2.65 Nom de la liste de distribution rapporteur (reporting-DL-name)

Cet attribut général contient l'argument *reporting-DL-name* (nom de la liste de distribution rapporteur) de l'opération abstraite Report-delivery (remise de rapport). Il identifie la liste de distribution qui a retransmis le rapport au propriétaire de cette liste de distribution; voir 8.3.1.2.1.4 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-reporting-DL-name ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ReportingDLName,
  EQUALITY MATCHING-RULE     oRNameMatch, -- règle non définie dans les contextes d'application
                               1988 --
  OTHER MATCHING-RULES      {oRNameElementsMatch | oRNameSubstringElementsMatch |
                             oRNameSingleElementMatch},
  NUMERATION                 single-valued,
  ID                          id-att-reporting-DL-name }

```

11.2.66 Certificat de l'agent MTA rapporteur (reporting-MTA-certificate)

Cet attribut général contient l'argument *reporting-MTA-certificate* (certificat de l'agent MTA rapporteur) de l'opération abstraite Report-delivery (remise de rapport). Il contient le certificat de l'agent MTA qui a créé le rapport; voir 8.3.1.2.1.12 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-reporting-MTA-certificate ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ReportingMTACertificate,
  NUMERATION                 single-valued,
  ID                          id-att-reporting-MTA-certificate }

```

11.2.67 Vérification d'authentification de l'origine du rapport (report-origin-authentication-check)

Cet attribut général contient l'argument *report-origin-authentication-check* (vérification d'authentification de l'origine du rapport) de l'opération abstraite Report-delivery (remise de rapport). Il offre un moyen pour authentifier l'origine du rapport; voir 8.3.1.2.1.13 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```

mt-report-origin-authentication-check ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ReportOriginAuthenticationCheck,
  NUMERATION                 single-valued,
  ID                          id-att-report-origin-authentication-check }

```

11.2.68 Etat d'extraction (retrieval-status)

Cet attribut général indique l'état d'extraction d'une entrée; voir 6.3.8. Si l'opération abstraite Modify (modification) est disponible, l'utilisateur MS sera capable de modifier l'attribut, à condition que sa valeur existante soit *listed* (listé) et que la valeur de remplacement soit *processed* (traité). L'attribut n'est pas soumis à des modifications au moyen de l'action automatique Auto-modify (modification automatique). Dans une entrée de la classe Message-log (journal de messagerie), l'attribut reflète l'état d'extraction de l'entrée correspondante dans la classe d'entrées Stored-message

(message enregistré). Dans une entrée de la classe Auto-action-log (journal d'actions automatiques), l'attribut indique l'état d'extraction de l'entrée elle-même (et non celui de l'entrée de l'objet de l'action automatique). L'attribut est créé et conservé par la mémoire de messages.

```
ms-retrieval-status ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      RetrievalStatus,
  EQUALITY MATCHING-RULE    integerMatch,
  NUMERATION                 single-valued,
  ID                          id-att-retrieval-status }
```

11.2.69 Classification de sécurité (security-classification)

Cet attribut général comprend la composante *security-classification* (classification de sécurité) de l'attribut message-security-label (étiquette de sécurité du message). Il est défini comme un attribut distinct afin de pouvoir être utilisé dans l'opération abstraite Summarize (récapitulation); voir 8.5.9 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-security-classification ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      SecurityClassification,
  EQUALITY MATCHING-RULE    integerMatch,
  NUMERATION                 single-valued,
  ID                          id-att-security-classification }
```

11.2.70 Numéro de séquence (sequence-number)

Cet attribut général identifie l'entrée contenant l'attribut. Il est attribué par la mémoire de messages lorsque l'entrée est créée; voir 6.3.2.

```
ms-sequence-number ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      SequenceNumber,
  EQUALITY MATCHING-RULE    integerMatch,
  ORDERING MATCHING-RULE    integerOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-sequence-number }
```

11.2.71 Période d'enregistrement (storage-period)

Cet attribut général indique la durée, en secondes, prévue par l'utilisateur MS comme étant nécessaire pour l'enregistrement de l'entrée, par rapport à son heure de création. Une entrée s'expose à la suppression par l'action automatique Auto-delete (suppression automatique) lorsque sa période d'enregistrement expire; voir 13.4. Une mémoire de messages qui prend en charge cet attribut général doit prendre en charge sa création et sa modification par l'opération abstraite Modify (modification) et par l'action automatique Auto-modify (modification automatique); elle doit également prendre en charge l'attribut général storage-time (heure d'enregistrement) auquel cet attribut général est lié sémantiquement. Une valeur donnée de l'attribut storage-period est égale à la valeur de l'heure d'enregistrement correspondante moins l'heure de création, exprimée en secondes. La création, la modification, ou la suppression de l'attribut général storage-period se reflète dans la création, la modification, ou la suppression de l'attribut général storage-time; c'est-à-dire que le lien sémantique entre les deux est toujours préservé.

```
ms-storage-period ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      StoragePeriod,
  EQUALITY MATCHING-RULE    integerMatch,
  ORDERING MATCHING-RULE    integerOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-storage-period }
```

StoragePeriod ::= INTEGER -- secondes --

NOTE – Il est prévu d'utiliser cet attribut avec l'action automatique Auto-modify (modification automatique), pour permettre à l'utilisateur MS d'affecter aux entrées sélectionnées une heure d'enregistrement égale à l'heure de création de l'entrée plus une durée fixée. Il est impossible d'utiliser directement l'heure d'enregistrement étant donné que cet attribut contient une heure absolue.

11.2.72 Heure d'enregistrement (storage-time)

Cet attribut général indique la date et l'heure après lesquelles l'utilisateur MS estime que l'enregistrement d'une entrée ne sera plus requis. Une entrée s'expose à la suppression par l'action automatique Auto-delete (suppression automatique) lorsque l'heure de son enregistrement est atteinte; voir 13.4. Une mémoire de messages qui prend en charge cet attribut général doit prendre en charge sa création et sa modification par l'opération abstraite Modify (modification) et par l'action automatique Auto-modify (modification automatique); elle doit également prendre en charge l'attribut général storage-period (période d'enregistrement) auquel cet attribut général est lié sémantiquement. La création, la modification, ou la suppression de l'attribut général storage-time se reflète dans la création, la modification, ou la suppression de

l'attribut général *storage-period* (période d'enregistrement); c'est-à-dire que le lien sémantique entre les deux est toujours préservé.

```
ms-storage-time ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      StorageTime,
  EQUALITY MATCHING-RULE    uTCTimeMatch,
  ORDERING MATCHING-RULE   uTCTimeOrderingMatch,
  NUMERATION                 single-valued,
  ID                          id-att-storage-time }
```

StorageTime ::= UTCTime

11.2.73 Identificateur de dépôt du sujet (subject-submission-identifiant)

Cet attribut général contient l'argument *subject-submission-identifiant* (identificateur de dépôt du sujet) de l'opération abstraite Report-delivery (remise de rapport). Il indique l'identificateur de dépôt du message *message-submission-identifiant* ou l'identificateur de dépôt du message d'essai *probe-submission-identifiant* du sujet du rapport; voir 8.3.1.2.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-subject-submission-identifiant ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      SubjectSubmissionIdentifier,
  EQUALITY MATCHING-RULE    mTSIdentifierMatch, -- règle non définie dans les contextes d'application
                                                                    1988 --
  NUMERATION                 single-valued,
  ID                          id-att-subject-submission-identifiant }
```

11.2.74 Nom de ce destinataire (this-recipient-name)

Cet attribut général contient l'argument *this-recipient-name* (nom de ce destinataire) de l'opération abstraite Message-delivery (remise de message) et identifie l'utilisateur MS; voir 8.3.1.1.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-this-recipient-name ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      ORName,
  EQUALITY MATCHING-RULE    oRNameMatch,
  OTHER MATCHING-RULES     {oRNameElementsMatch | oRNameSubstringElementsMatch |
                                                                    oRNameSingleElementMatch},
  NUMERATION                 single-valued,
  ID                          id-att-this-recipient-name }
```

11.2.75 Information de trace (trace-information)

Cet attribut général, qui est multi-valué, fournit des détails sur les mesures concernant le message (ou le message d'essai ou le rapport) prises par chaque domaine de gestion par lequel il transite lors de son transfert à travers le système MTS. Il est créé à partir des attributs Message-delivery-envelope (enveloppe de remise du message) et Report-delivery-envelope (enveloppe de remise du rapport); voir 12.2.1.1.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4.

```
mt-trace-information ATTRIBUTE ::= {
  WITH ATTRIBUTE-SYNTAX      TraceInformationElement,
  NUMERATION                 multi-valued,
  ID                          id-att-trace-information }
```

11.3 Ensemble d'objets informationnels (Attribute-table)

L'ensemble d'objets informationnels **Attribute-table** (tableau des attributs) est utilisé comme un ensemble de contraintes dans la présente Définition de service quand des champs liés à la classe d'objets informationnels ATTRIBUTE (attribut) sont référencés dans la syntaxe abstraite MS. Cet ensemble est composé de deux ensembles d'objets:

```
AttributeTable ATTRIBUTE ::= {
  GeneralAttributes | ContentSpecificAttributes }
```

L'ensemble d'objets informationnels **General-attributes** (attributs généraux) contient les attributs généraux définis dans la présente Définition de service. Il est défini dans l'Annexe C. La réalisation d'une mémoire de messages n'est pas contrainte à ne prendre en charge que les attributs généraux présents dans l'ensemble d'objets General-attributes.

L'ensemble d'objets informationnels **Content-specific-attributes** (attributs propres au contenu) est un ensemble vide et extensible. Quand une mémoire de messages offre de prendre en charge un type de contenu donné, les types d'attributs associés à ce type de contenu seront considérés comme des objets appartenant à l'ensemble d'objets Content-specific-attributes.

```
ContentSpecificAttributes ATTRIBUTE ::= { ... }
```

11.4 Création des attributs généraux

Ce paragraphe décrit la manière dont sont créés les attributs généraux. Les informations sont récapitulées dans le Tableau 4; les descriptions définitives des attributs sont données au 11.2. On trouvera une description des classifications utilisées au 5.3.

11.5 Abonnement aux types d'attributs

L'abonnement aux types d'attributs relève des autorités locales. Si l'abonnement à un type d'attribut est modifié de sorte qu'un plus petit nombre d'attributs sont soumis à un abonnement, l'utilisateur MS peut continuer à avoir accès à tous les attributs qui étaient soumis à un abonnement au départ pour les messages présents dans la mémoire de messages au moment où l'abonnement a été modifié. Le traitement des attributs pour lesquels l'abonnement a été annulé relève des autorités locales. De même, lors de l'abonnement à un nouvel attribut, l'utilisateur MS peut ne pas avoir accès à cet attribut pour les messages présents dans la mémoire de messages au moment où l'abonnement a été modifié.

Tableau 4 – Création des types d'attributs généraux

Nom du type d'attribut		Mono-/multi- valué	Paramètre source	Source créée par	Règles de création
AC-correlated-report-list	liste des rapports corrélés AC	M	–	MS	Une valeur est créée pour chaque destinataire spécifié au départ, identifiant les champs de remise de rapport par destinataire qui ont été corrélés avec ce destinataire.
AC-report-summary	récapitulation des rapports AC	M	–	MS	Les valeurs initiales sont les valeurs de l'attribut originator-report-request. Elles sont mises à jour à la réception de rapports. Une valeur est créée pour chaque destinataire spécifié au départ.
AC-uncorrelated-report-list	liste des rapports non corrélés AC	M	–	MS	Une valeur est créée pour chaque champ de remise de rapport par destinataire qui ne peut pas être corrélé avec un destinataire spécifié au départ.
Auto-action-error	erreur d'action automatique	S	–	MS	La valeur d'attribut est la valeur de l'erreur créée comme résultat de l'exécution de l'action automatique.
Auto-action-registration-identifiant	identificateur de consignation d'action automatique	S	–	MS	La valeur d'attribut est celle de l'identificateur de consignation d'une action automatique consignée exécutée par la mémoire de messages.
Auto-action-subject	objet de l'action automatique	S	–	MS	La valeur d'attribut est celle du numéro de séquence d'une entrée à laquelle a été appliquée une action automatique.
Auto-action-type	type d'action automatique	S	–	MS	La valeur d'attribut est le type d'une action automatique consignée exécutée par la mémoire de messages.
Child-sequence-numbers	numéros de séquence d'entrée fille	M	–	MS	Une valeur est créée pour chaque entrée fille appartenant à une entrée mère.
Content	contenu	S	content returned-content	Ms, Md Rd	La valeur d'attribut dans l'entrée de message remis ou déposé est la valeur du paramètre source. La valeur d'attribut dans l'entrée fille de contenu renvoyé de l'entrée de rapport remis est la valeur du paramètre source.
Content-confidentiality-algorithm-identifiant	identificateur d'algorithme de confidentialité du contenu	S	content-confidentiality-algorithm-identifiant	Ms, Md	La valeur d'attribut est la valeur du paramètre source.
Content-correlator	corrélateur de contenu	S	content-correlator	Ms, Ps, Rd	La valeur d'attribut est la valeur du paramètre source.
Content-identifiant	identificateur de contenu	S	content-identifiant	Ms, Ps, Md, Rd	La valeur d'attribut est la valeur du paramètre source.
Content-integrity-check	vérification d'intégrité du contenu	S	content-integrity-check	Md	La valeur d'attribut est la valeur du paramètre source.

Tableau 4 (suite) – Création des types d'attributs généraux

Nom du type d'attribut		Mono-/multi- valué	Paramètre source	Source créée par	Règles de création
Content-length	longueur du contenu	S	–	MS	Il s'agit de la taille (approximative) du contenu en octets.
			content-length	Ps	La valeur d'attribut est la valeur du paramètre source.
Content-returned	contenu renvoyé	S	–	MS	La valeur est mise à vrai si le paramètre returned-content est présent dans une opération Report-delivery et à faux s'il est absent.
Content-type	type de contenu	S	content-type	Ms, Ps, Md, Rd	Si cet attribut est représenté par un identificateur d'objet, il s'agit de la valeur du paramètre source. S'il est représenté par un nombre entier, la valeur est convertie à l'identificateur d'objet correspondant.
Conversion-with-loss-prohibited	conversion avec perte interdite	S	conversion-with-loss-prohibited	Ms, Ps, Md	La valeur d'attribut est la valeur du paramètre source.
Converted-EITs	types d'information codée convertis	M	converted-encoded-information-types	Md	Une valeur correspondante est créée pour chaque bit mis à 1 dans la composante built-in-encoded-information-types (types d'information codée intégrés) et à partir de chaque type d'information codée étendu présent dans le paramètre converted-encoded-information-types.
Creation-time	heure de création	S	–	MS	Heure de création de l'entrée.
Deferred-delivery-cancellation-time	heure d'annulation de remise différée	S	–	MS	Heure à laquelle l'opération abstraite Cancel-deferred-delivery a été exécutée.
Deferred-delivery-time	heure de remise différée	S	deferred-delivery-time	Ms	La valeur d'attribut est la valeur du paramètre source.
Deletion-time	heure de suppression	S	–	MS	Attribut créé lorsque l'entrée à laquelle correspond cette entrée de la classe Message-log est supprimée.
Delivered-EITs	types d'information codée remis	M	converted-encoded-information-types, si présent; original-encoded-information-types sinon	MS	Converted-EITs si le paramètre converted-encoded-information-types est présent; Original-EITs sinon.
Delivery-flags	fanions de remise	S	delivery-flags	Md	La valeur d'attribut est la valeur du paramètre source. S'il n'y a pas de fanion de remise dans l'opération Md, une valeur par défaut est créée sans fanion positionné.
DL-expansion-history	chronologie du développement de la liste de distribution	M	dl-expansion-history	Md	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
			originator-and-dl-expansion-history	Rd	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
DL-expansion-prohibited	interdiction du développement de la liste de distribution	S	dl-expansion-prohibited	Ms, Ps	La valeur d'attribut est la valeur du paramètre source.
Entry-type	type d'entrée	S	Argument Md	MS	La valeur est «delivered-message».
			Argument Rd	MS	La valeur est «delivered-report». Si un contenu renvoyé est présent, une entrée fille qui contient le contenu renvoyé est créée et on lui donne le type d'entrée «returned-content».
			Argument Ms	MS	La valeur est «submitted-message».
			Argument Ps	MS	La valeur est «submitted-probe».
			Argument Ms	MS	La valeur est «draft-message».
				MS	La valeur est «auto-action-event».

Tableau 4 (suite) – Création des types d'attributs généraux

Nom du type d'attribut		Mono- /multi- valué	Paramètre source	Source créée par	Règles de création
Internal-trace-information	information de trace interne	M	internal-trace-information	Md Rd	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
Latest-delivery-time	heure limite de remise	S	latest-delivery-time	Ms	La valeur d'attribut est la valeur du paramètre source.
Marked-for-deletion	marqué pour être supprimé	S	Argument Modify, Argument Auto-modify	Mod, Amod	Attribut créé par l'opération abstraite Modify ou l'action automatique Auto-modify.
Message-delivery-envelope	enveloppe de remise de message	S	MessageDelivery Envelope	Md	La valeur d'attribut est la valeur du paramètre source.
Message-delivery-time	heure de remise de message	S	message-delivery-time	Md	La valeur d'attribut est la valeur du paramètre source.
Message-group-name	nom de groupe de messages	M	Argument Modify, Argument Auto-modify	Mod, Amod	Attribut créé par l'opération abstraite Modify ou l'action automatique Auto-modify.
Message-identifiant	identificateur de message	S	message-submission- identifiant, probe-submission- identifiant, message-delivery- identifiant	Ms, Ps, Md	La valeur d'attribut est la valeur du paramètre source.
Message-notes	notes de message	M	Argument Modify, Argument Auto-modify	Mod, Amod	Attribut créé par l'opération abstraite Modify ou l'action automatique Auto-modify.
Message-origin-authentication-check	vérification d'authentification de l'origine du message	S	message-origin-authentication-check	Ms, Md	La valeur d'attribut est la valeur du paramètre source.
Message-security-label	étiquette de sécurité du message	S	message-security-label	Ms, Ps, Md, Rd	La valeur d'attribut est la valeur du paramètre source.
Message-submission-envelope	enveloppe de dépôt du message	S	Message-Submission- Envelope	Ms	La valeur d'attribut est la valeur du paramètre source.
Message-submission-time	heure de dépôt du message	S	message-submission-time, probe-submission-time	Ms, Md Ps	La valeur d'attribut est la valeur du paramètre source. La valeur d'attribut est la valeur du paramètre source.
Message-token	jeton de message	S	message-token	Md	La valeur d'attribut est la valeur du paramètre source.
MS-originated	origine MS	S	–	MS	Positionner à <i>vrai</i> si la mémoire de messages a expédié ce message dans l'exécution d'une action automatique.
MS-submission-error	erreur de dépôt MS	S	MS-message-submission error, MS-probe- submission error	MS	Attribut créé par l'erreur ERROR renvoyée par les opérations abstraites MS-message-submission ou MS-probe-submission.
Original-EITs	types d'information codée d'origine	M	original-encoded- information-types	Ms, Ps, Md, Rd	Une valeur correspondante est créée pour chaque bit mis à 1 dans la composante built-in-encoded-information-types (types d'information codée intégrés) et à partir de chaque type d'information codée étendu présent dans le paramètre original-encoded-information-types.
Originally-intended-recipient-name	nom du destinataire prévu au départ	S	originally-intended- recipient-name	Md	La valeur d'attribut est la valeur du paramètre source.
Originating-MTA-certificate	certificat de l'agent MTA expéditeur	S	originating-mta- certificate	Ms	La valeur d'attribut est la valeur du paramètre source.
Originator-certificate	certificat de l'expéditeur	S	originator-certificate	Ms, Ps, Md	La valeur d'attribut est la valeur du paramètre source.
Originator-name	nom de l'expéditeur	S	originator-name	Ms, Ps, Md	La valeur d'attribut est la valeur du paramètre source.

Tableau 4 (suite) – Création des types d'attributs généraux

Nom du type d'attribut		Mono-/multi- valué	Paramètre source	Source créée par	Règles de création
Originator-report-request	demande de rapport par l'expéditeur	M	originator-report-request	Ms, Ps	La valeur d'attribut est la valeur du paramètre source.
Originator-return-address	adresse de retour à l'expéditeur	S	originator-return-address	Ms, Md	La valeur d'attribut est la valeur du paramètre source.
Other-recipient-names	noms d'autres destinataires	M	other-recipient-names	Md	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
Parent-sequence-number	numéro de séquence d'entrée mère	S	–	MS	Lors de la création d'une entrée fille, cet attribut est créé avec la valeur du numéro de séquence de l'entrée mère correspondante.
Per-message-indicators	indicateurs par message	S	per-message-indicators	Ms, Ps	La valeur d'attribut est la valeur du paramètre source. S'il ne figure pas dans les opérations Ms ou Ps, une chaîne de bits zéro est créée par défaut.
Per-recipient-message-submission-fields	champs de dépôt de message par destinataire	M	PerRecipientMessageSubmissionFields	Ms	La valeur d'attribut est la valeur du paramètre source.
Per-recipient-probe-submission-fields	champs de dépôt de message d'essai par destinataire	M	PerRecipientProbeSubmissionFields	Ps	La valeur d'attribut est la valeur du paramètre source.
Per-recipient-report-delivery-fields	champs de remise de rapport par destinataire	M	per-recipient-fields	Rd	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
Priority	priorité	S	priority	Ms, Md	La valeur d'attribut est la valeur du paramètre source.
Probe-origin-authentication-check	vérification d'authentification de l'origine du message d'essai	S	probe-origin-authentication-check	Ps	La valeur d'attribut est la valeur du paramètre source.
Probe-submission-envelope	enveloppe de dépôt du message d'essai	S	ProbeSubmissionEnvelope	Ps	La valeur d'attribut est la valeur du paramètre source.
Proof-of-delivery-request	demande de preuve de remise	S	proof-of-delivery-request	Md	La valeur d'attribut est la valeur du paramètre source.
Proof-of-submission	preuve de dépôt	S	proof-of-submission	Ms	La valeur d'attribut est la valeur du paramètre source.
Recipient-names	noms des destinataires	M	recipient-name	Ms, Ps	La valeur d'attribut est la valeur du paramètre source.
Recipient-reassignment-prohibited	interdiction de réaffectation de destinataire	S	recipient-reassignment-prohibited	Ms, Ps	La valeur d'attribut est la valeur du paramètre source.
Redirection-history	chronologie du réacheminement	M	redirection-history	Md, Rd	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.
Report-delivery-envelope	enveloppe de remise de rapport	S	ReportDeliveryEnvelope	Rd	La valeur d'attribut est la valeur du paramètre source.
Reporting-DL-name	nom de la liste de distribution rapporteur	S	reporting-dl-name	Rd	La valeur d'attribut est la valeur du paramètre source.
Reporting-MTA-certificate	certificat de l'agent MTA rapporteur	S	reporting-MTA-certificate	Rd	La valeur d'attribut est la valeur du paramètre source.
Report-origin-authentication-check	vérification d'authentification de l'origine du rapport	S	report-origin-authentication-check	Rd	La valeur d'attribut est la valeur du paramètre source.
Retrieval-status	état d'extraction	S	–	MS	Attribut créé lors de la création de l'entrée; on lui affecte la valeur 'new' (nouveau).
Security-classification	classification de sécurité	S	security-classification	Md, Rd	La valeur d'attribut est la valeur du paramètre source.
Storage-period	période d'enregistrement	S	Argument Modify, Argument Auto-modify	Mod, Amod	Attribut créé par l'opération abstraite Modify ou par l'action automatique Auto-modify. L'attribut storage-time est créé en même temps.
Storage-time	heure d'enregistrement	S	Argument Modify, Argument Auto-modify	Mod, Amod	Attribut créé par l'opération abstraite Modify ou par l'action automatique Auto-modify. L'attribut storage-period est créé en même temps.

Tableau 4 (*fin*) – Création des types d'attributs généraux

Nom du type d'attribut		Mono- /multi- valué	Paramètre source	Source créée par	Règles de création
Sequence-number	numéro de séquence	S	–	MS	Lorsqu'elle crée une entrée, la mémoire de messages affecte une valeur unique à cet attribut par ordre croissant.
Subject-submission- identifier	identificateur de dépôt du sujet	S	subject-submission- identifier	Rd	La valeur d'attribut est la valeur du paramètre source.
This-recipient-name	nom de ce destinataire	S	this-recipient-name	Md	La valeur d'attribut est la valeur du paramètre source.
Trace-information	information de trace	M	trace-information	Md Rd	Une valeur correspondante est créée à partir de chaque composante de la SEQUENCE.

11.6 Types d'attributs généraux pouvant être modifiés

Parmi les types d'attributs généraux, seuls ceux qui sont énumérés ci-dessous peuvent être modifiés par l'opération abstraite Modify (modification), et, à l'exception de l'attribut retrieval-status (état d'extraction), par l'action automatique Auto-modify (modification automatique). A l'exception de l'attribut retrieval-status, quand une mémoire de messages prend en charge l'un de ces attributs, elle doit prendre en charge sa modification par l'opération abstraite Modify (modification).

- a) Marked-for-deletion (marqué pour être supprimé)
- b) Message-group-name (nom de groupe de messages)
- c) Message-notes (notes de message)
- d) Retrieval-status (état d'extraction)
- e) Storage-period (période d'enregistrement)
- f) Storage-time (heure d'enregistrement)

Des détails concernant les attributs propres au contenu qui peuvent être modifiés sont donnés dans la Spécification qui définit le type de contenu en question.

12 Règles de criblage générales

Les **règles de criblage générales** peuvent être appliquées aux attributs d'une entrée de n'importe quel type de contenu. D'autres règles de criblage, qui sont propres au contenu, sont définies dans la Spécification en question. Certaines règles sont basées sur les règles de criblage définies dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 et dans la Rec. UIT-T X.520 | ISO/CEI 9594-6. La classe d'objets informationnels MATCHING-RULE (règle de criblage) utilisée dans la présente Définition de service est décrite au 6.3.9.3.

12.1 Syntaxe MS-string (chaîne MS)

La syntaxe d'attribut de certains types d'attributs permet de représenter les mêmes informations sous des formes différentes:

- comme une chaîne numérique Numeric String ou une chaîne imprimable Printable String;
- comme une chaîne imprimable Printable String ou une chaîne de télétexte Teletex String.

Pour les attributs standard d'adresse OR, ces différentes formes de représentation sont énumérées dans le Tableau 9 de la Rec. UIT-T X.402 | ISO/CEI 10021-2. Il est pratique de définir une syntaxe ASN.1 pour spécifier ces formes optionnelles dans les assertions de règles de criblage.

```

MSString { INTEGER : maxSize } ::= CHOICE {
    printable          PrintableString (SIZE (1..maxSize)),
    teletex            TeletexString (SIZE (1..maxSize)),
    general            GeneralString (SIZE (1..maxSize)),
    universal          UniversalString (SIZE (1..maxSize)),
    bmp                BMPString (SIZE (1..maxSize)) }

```

Quand le type MS-string est utilisé comme syntaxe d'assertion dans une assertion de règle de criblage, le choix des types de chaîne est limité au choix représenté dans la syntaxe de l'attribut sous test.

Les contraintes sur l'ensemble des caractères de type Universal String (chaîne universelle) qui peuvent être présents dans une chaîne MS-string seront précisées dans les Spécifications qui définissent les types d'attribut dont la syntaxe d'attribut est Universal String (chaîne universelle). Ces types d'attribut ne sont pas définis dans la présente Définition de service.

12.2 Règles de criblage basées sur les chaînes

Dans les règles de criblage spécifiées dans 12.2.1 à 12.2.9, on considère que les espaces suivants ne sont pas significatifs:

- espaces de début (c'est-à-dire ceux qui précèdent le premier caractère imprimé);
- espaces de fin (c'est-à-dire ceux qui suivent le dernier caractère imprimé);
- plusieurs espaces internes consécutifs (ils sont équivalents à un seul caractère d'espacement).

Dans les règles de criblage auxquelles ces considérations s'appliquent, les chaînes à comparer le seront comme si les espaces non significatifs n'étaient présents dans aucune chaîne. On considère qu'une valeur qui ne contient que des caractères d'espacement contient un seul espace significatif.

Les règles suivantes s'appliquent également:

- le caractère de soulignement à chasse nulle sera considéré comme non significatif, de même que toutes les fonctions de commande sauf la fonction d'espacement et celles qui sont utilisées pour les procédures d'extension de code;
- le choix entre différents codages du même caractère sera considéré comme non significatif.

Lorsque les chaînes comparées possèdent des types de chaîne de caractères ASN.1 différents (par exemple une chaîne imprimable Printable String présentée et une chaîne de télétexte Teletext String enregistrée), la comparaison se déroule normalement tant que les caractères correspondants sont dans les deux ensembles de caractères; la comparaison est basée sur l'équivalence des caractères et on ne tient pas compte du choix du codage. Sinon la comparaison échoue.

12.2.1 Crible de chaînes MS

La règle **MS-string-match** (criblage de chaînes MS) compare une chaîne présentée aux valeurs d'attribut de type MS-string pour déterminer si elles sont égales, sans tenir compte du fait que leurs caractères soient des minuscules ou des majuscules (par exemple, comparaison entre «Dundee» et «DUNDEE»).

NOTE – Dans certaines écritures, la distinction entre minuscules et majuscules n'existe pas.

```
mSStringMatch MATCHING-RULE ::= {
  SYNTAX          MSString {ub-msstring-match}
  ID              id-mr-msstring-match }
```

La règle renvoie *vrai* si et seulement si les chaînes ont le même nombre de caractères et si les caractères correspondants sont identiques (on ne tient pas compte de la capitalisation). Quand le type d'attribut autorise le choix entre différents types de chaînes de caractères ASN.1, la comparaison est basée sur l'équivalence des caractères et on ne tient pas compte du choix du codage.

12.2.2 Crible d'ordonnement de chaînes MS

La règle **MS-string-ordering-match** (crible d'ordonnement de chaînes MS) compare l'ordre de collation d'une chaîne présentée par rapport aux valeurs d'attribut de type MS-string, en ne tenant pas compte de la capitalisation des lettres.

```
mSStringOrderingMatch MATCHING-RULE ::= {
  SYNTAX          MSString {ub-msstring-match}
  ID              id-mr-msstring-ordering-match }
```

La règle renvoie *vrai* si et seulement si une certaine valeur de l'attribut vaut 'moins' ou apparaît plus tôt que la valeur présentée, quand les chaînes sont comparées en utilisant l'ordre normal de collation pour leur syntaxe après avoir remplacé les minuscules des deux chaînes par les majuscules correspondantes.

NOTES

- 1 Par exemple, la règle renvoie *vrai* quand la valeur présentée «Falkirk» est comparée à un attribut contenant les valeurs «Glasgow» et «Edinburgh», étant donné que la valeur «Edinburgh» vaut 'moins' que la valeur présentée.
- 2 La définition de l'ordre normal de collation peut faire l'objet d'une décision nationale.

12.2.3 Crible de sous-chaînes MS

La règle **MS-substrings-match** (crible de sous-chaînes MS) détermine si une valeur présentée est une sous-chaîne d'une certaine valeur d'un attribut de type MS-string, en ne tenant pas compte de la capitalisation des caractères.

```
mSSubstringsMatch MATCHING-RULE ::= {
  SYNTAX  SubstringAssertion
  ID      id-mr-ms-substrings-match }

SubstringAssertion ::= SEQUENCE OF CHOICE {
  initial  [0] MSString {ub-msstring-match},
  any      [1] MSString {ub-msstring-match},
  final    [2] MSString {ub-msstring-match} }
-- au plus une composante initiale et une composante finale --
```

Pour comparer la valeur d'une sous-chaîne, les sous-chaînes spécifiées doivent apparaître dans l'ordre de l'assertion. Les sous-chaînes ne doivent pas se chevaucher, et peuvent (mais cela n'est pas indispensable) être séparées des extrémités de la valeur d'attribut et les unes des autres par zéro, un ou plusieurs éléments de chaîne.

Si la composante **initial** est présente, la sous-chaîne doit correspondre à la sous-chaîne initiale de la valeur; si la composante **final** est présente, la sous-chaîne doit correspondre à la sous-chaîne finale de la valeur; si la composante **any** (n'importe) est présente, la sous-chaîne doit correspondre à n'importe quelle sous-chaîne de la valeur. Chaque sous-chaîne est comparée conformément à la règle MS-string-match.

Pour qu'une composante d'assertion de sous-chaîne corresponde à une portion de la valeur d'attribut, les caractères homologues doivent être identiques, compte non tenu de leur capitalisation. Quand le type d'attribut permet de choisir entre différents types de chaînes de caractères ASN.1, la comparaison est basée sur l'équivalence des caractères et on ne tient pas compte du choix du codage.

NOTE – Par exemple, la règle renvoie *vrai* quand une valeur présentée contenant «mes» comme composante initial, «age» comme composante any et «ore» comme composante final est comparée à un attribut contenant la valeur «Message Store».

12.2.4 Crible de sous-chaînes individuelles MS

La règle **MS-single-substring-match** (crible de sous-chaînes individuelles MS) détermine si une chaîne présentée est une sous-chaîne individuelle d'une certaine valeur d'un attribut de type MS-string, en ne tenant pas compte de la capitalisation des caractères.

```
mSSingleSubstringMatch MATCHING-RULE ::= {
  SYNTAX  MSString {ub-msstring-match}
  ID      id-mr-ms-single-substring-match }
```

La règle est identique à la règle MS-substrings-match, sauf que la valeur présentée est traitée comme l'unique composante **any** (n'importe) de l'assertion de sous-chaînes; les composantes **initial** et **final** sont absentes.

NOTE – Par exemple, la règle renvoie *vrai* quand la valeur présentée «Age» est comparée à un attribut contenant la valeur «Message Store».

12.2.5 Crible de chaînes MS sensible à la capitalisation

La règle **MS-string-case-sensitive-match** (crible de chaînes MS sensible à la capitalisation) compare une chaîne présentée aux valeurs d'attribut de type MS-string.

```
mSSStringCaseSensitiveMatch MATCHING-RULE ::= {
  SYNTAX  MSString {ub-msstring-match}
  ID      id-mr-msstring-case-sensitive-match }
```

La règle est identique à la règle MS-string-match sauf qu'il est tenu compte de la capitalisation.

NOTE – Par exemple, la règle renvoie *vrai* pour la valeur présentée «CaSe» si et seulement si cette valeur est comparée à un attribut contenant la valeur «CaSe».

12.2.6 Crible de listes de chaînes MS

La règle **MS-string-list-match** (crible de listes de chaînes MS) compare une séquence présentée de chaînes aux valeurs d'attribut qui sont des séquences de chaînes MS-string pour déterminer si elles sont égales, en ne tenant pas compte de la capitalisation des caractères.

```
mSSStringListMatch MATCHING-RULE ::= {
  SYNTAX  SEQUENCE OF MSString {ub-msstring-match}
  ID      id-mr-msstring-list-match }
```

La règle renvoie *vrai* si et seulement si le nombre de chaînes est le même dans chaque séquence, et si les chaînes homologues se correspondent pour la règle de criblage MS-string-match.

NOTE – Par exemple, la règle renvoie *vrai* quand la valeur présentée «ms projects+milestones+annoncements» est comparée à un attribut contenant la valeur «MS Projects+Milestones+Announcements» (où «+» sépare les chaînes dans la séquence).

12.2.7 Crible d'éléments de listes de chaînes MS

La règle **MS-string-list-elements-match** (crible d'éléments de listes de chaînes MS) détermine si une séquence présentée de chaînes est un sous-ensemble du début de la séquence de chaînes présente dans une certaine valeur d'un attribut contenant des séquences de chaînes de type MS-string, en ne tenant pas compte de la capitalisation des caractères.

```
mSStringListElementsMatch MATCHING-RULE ::= {
  SYNTAX SEQUENCE OF MSSString {ub-msstring-match}
  ID      id-mr-msstring-list-elements-match }
```

La règle est identique à la règle MS-string-list-match sauf que les chaînes sont comparées dans l'ordre présenté dans la séquence (en commençant par la première chaîne dans chaque séquence), et la règle renvoie *vrai* quand toutes les chaînes de la séquence présentée correspondent (c'est-à-dire que le nombre de chaînes de la séquence présentée doit être inférieur ou égal au nombre de chaînes de la valeur enregistrée).

NOTE – Par exemple, la règle renvoie *vrai* quand la valeur présentée «ms projects+milestones» est comparée à un attribut contenant la valeur «MS Projects+Milestones+Announcements» (où «+» sépare les chaînes dans la séquence).

12.2.8 Crible de listes de sous-chaînes individuelles MS

La règle **MS-single-substring-list-match** (crible de listes de sous-chaînes individuelles MS) détermine si chaque chaîne d'une séquence présentée est une sous-chaîne individuelle de chaque chaîne homologue dans une séquence qui constitue une certaine valeur d'un attribut contenant des séquences de chaînes de type MS-string, en ne tenant pas compte de la capitalisation des caractères.

```
mSSingleSubstringListMatch MATCHING-RULE ::= {
  SYNTAX SEQUENCE OF MSSString {ub-msstring-match}
  ID      id-mr-ms-single-substring-list-match }
```

La règle renvoie *vrai* si et seulement si le nombre de chaînes est le même dans chaque séquence, et si les chaînes homologues se correspondent pour la règle MS-single-substring-match.

NOTE – Par exemple, la règle renvoie *vrai* quand la valeur présentée «projects+miles+ounce» est comparée à un attribut contenant la valeur «MS Projects+Milestones+Announcements» (où «+» sépare les chaînes dans la séquence).

12.2.9 Crible d'éléments de listes de sous-chaînes individuelles MS

La règle **MS-single-substring-list-elements-match** (crible d'éléments de listes de sous-chaînes individuelles MS) détermine si une séquence présentée de sous-chaînes individuelles est un sous-ensemble de la séquence de chaînes présente dans une certaine valeur d'un attribut contenant des séquences de chaînes de type MS-string, en ne tenant pas compte de la capitalisation des caractères.

```
mSSingleSubstringListElementsMatch MATCHING-RULE ::= {
  SYNTAX SEQUENCE OF MSSString {ub-msstring-match}
  ID      id-mr-ms-single-substring-list-elements-match }
```

La règle est identique à la règle MS-single-substring-list-match sauf que les chaînes sont comparées dans l'ordre présenté dans la séquence (en commençant par la première chaîne dans chaque séquence), et la règle renvoie *vrai* quand toutes les chaînes de la séquence présentée correspondent (c'est-à-dire que le nombre de chaînes de la séquence présentée doit être inférieur ou égal au nombre de chaînes de la valeur enregistrée).

NOTE – Par exemple, la règle renvoie *vrai* quand la valeur présentée «ms+stones» est comparée à un attribut contenant la valeur «MS Projects+Milestones+Announcements» (où «+» sépare les chaînes dans la séquence).

12.3 Règles de criblage syntaxiques

La présente Définition de service utilise les règles de criblage syntaxiques suivantes définies dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 et dans la Rec. UIT-T X.520 | ISO/CEI 9594-6: Bit String Match, Object Identifier Match, Distinguished Name Match, Boolean Match, Integer Match, Integer Ordering Match, Octet String Match, UTC Time Match, UTC Time Ordering Match et Presentation Address Match. En plus de ces règles, une règle de criblage value-count-matching est définie.

NOTE 1 – La règle Integer Match définie dans la Rec. UIT-T X.520 | ISO/CEI 9594-6 est également appliquée ici aux types énumérés.

La règle **value-count-match** (crible de décomptes de valeurs) détermine si le nombre de valeurs présentes dans un attribut d'une entrée est égal à un entier présenté.

```
valueCountMatch MATCHING-RULE ::= {
  SYNTAX  INTEGER (1..ub-attribute-values)
  ID      id-mr-value-count-match }
```

La règle renvoie *vrai* si et seulement si l'attribut désigné contient un nombre de valeurs égal à l'entier présenté.

NOTE 2 – Cette règle pourrait être utilisée dans un filtre qui sélectionne les entrées qui appartiennent à un groupe de messages donné pour identifier les entrées qui n'appartiennent qu'à ce groupe de messages.

12.4 Règles de criblage pour les attributs complexes de la mémoire de messages

Les composantes d'une adresse OR sont définies à l'article 18 de la Rec. UIT-T X.402 | ISO/CEI 10021-2 comme des types d'attribut standard **standard attribute types** et comme des types d'attributs définis par domaine **domain defined attribute types** [mais elles ne sont pas définies de façon formelle comme des instances de la classe d'objets informationnels ATTRIBUTE (attribut)]. De nombreux types d'attribut standard sont des types de données complexes [par exemple personal-name (nom personnel)] et ne conviennent pas toujours à la comparaison en tant qu'unités complètes. Pour cette raison, les types d'attribut standard sont éclatés en leurs composantes primitives qu'on appelle **éléments d'adresse OR** dans la présente Définition de service.

NOTE – Ainsi le type d'attribut standard *personal-name* comprend les quatre éléments d'adresse OR *surname* (nom de famille), *given-name* (prénom), *initials* (initiales), et *generation-qualifier* (indication de génération).

12.4.1 Crible d'adresse OR

La règle **OR-address-match** (crible d'adresse OR) compare une valeur présentée aux valeurs d'attribut de type OR-address (adresse OR) pour déterminer si elles sont égales.

```
oRAddressMatch MATCHING-RULE ::= {
  SYNTAX  ORAddress
  ID      id-mr-oraddress-match }
```

La règle renvoie *vrai* si la valeur présentée et au moins une valeur de l'attribut enregistré contiennent chacune le même nombre d'éléments d'adresse OR du même type, et si les éléments homologues se correspondent. Sinon la comparaison échoue.

La comparaison des éléments est réalisée conformément aux règles suivantes, qui reflètent les règles d'équivalence des listes d'attributs définies dans la Rec. UIT-T X.402 | ISO/CEI 10021-2:

- a) Tous les éléments sont comparés en utilisant la règle MS-string-match, à l'exception:
 - 1) du type de terminal, qui est comparé par la règle Integer Match;
 - 2) de la forme étendue de l'adresse réseau. L'adresse E163-4-address est comparée par la règle MS-string-list-match. L'adresse PSAP-address est comparée en utilisant la règle Presentation Address Match.
- b) Quand la valeur d'un élément peut être une chaîne numérique ou une chaîne imprimable équivalente, chaque forme de la valeur présentée doit correspondre à chaque forme de la valeur enregistrée.

NOTE 1 – Pour le nom du domaine d'administration, le nom du domaine privé et le code postal, la même valeur numérique peut être représentée comme une chaîne numérique ou imprimable.

- c) La règle d'équivalence entre les formes X.121 et ISO 3166 de l'élément country-name (nom du pays) doit s'appliquer (voir Rec. UIT-T X.402 | ISO/CEI 10021-2).

NOTE 2 – Au niveau local, lorsque la forme X.121 attribue plus d'un numéro à un pays, on peut considérer ces numéros comme équivalents.

- d) Quand la valeur d'un élément peut être une chaîne imprimable ou une chaîne de télétexte équivalente, ou les deux, chaque forme de la valeur présentée doit correspondre à chaque forme de la valeur enregistrée. Quand les deux formes sont présentes (dans la valeur présentée, ou dans la valeur enregistrée, ou dans les deux), la comparaison de l'élément aboutit si la comparaison d'une forme au moins aboutit.

NOTE 3 – Par exemple, l'élément organization-name (nom d'organisation) peut être présent comme attribut standard intégré (chaîne imprimable), ou comme attribut d'extension (chaîne de télétexte), ou comme les deux. De même, l'élément common-name (nom commun) peut être présent comme l'un, l'autre ou les deux attributs d'extension: common-name (chaîne imprimable), et teletex-common-name (chaîne de télétexte).

- e) Les composantes type et valeur d'un élément domain-defined-attribute (attribut défini par domaine) peuvent être un couple de valeurs Printable String (chaîne imprimable) ou Teletex String (chaîne de télételex) équivalentes; chaque forme du couple de valeurs présentées doit correspondre à chaque forme du couple de valeurs enregistrées.
- f) L'ordre des éléments n'est pas signifiant, à l'exception des éléments dérivés des noms d'unités organisationnelles, qui doivent apparaître dans le même ordre dans la valeur présentée et dans la valeur enregistrée.

NOTE 4 – Au niveau local, quand une mémoire de messages a une connaissance propre de certains attributs définis par domaine, elle peut appliquer des règles supplémentaires qui tiennent compte des signes de ponctuation, de la capitalisation, ou de l'ordre relatif de ces attributs définis par domaine.

NOTE 5 – Un domaine de gestion peut imposer d'autres règles de criblage basées sur des règles d'équivalence locales, aux attributs qu'il affecte à ses propres utilisateurs et listes de distribution. La mémoire de messages peut utiliser des informations locales à propos de l'équivalence des valeurs d'attribut ou des combinaisons de valeurs pour déterminer si une adresse OR fournie identifie de façon non ambiguë une adresse OR enregistrée.

12.4.2 Crible d'éléments d'adresse OR

La règle **OR-address-elements-match** (crible d'éléments d'adresse OR) détermine si une valeur présentée est un sous-ensemble des éléments présents dans une certaine valeur d'un attribut de type OR-address (adresse OR).

```
oRAddressElementsMatch MATCHING-RULE ::= {
  SYNTAX      ORAddress
  ID          id-mr-oraddress-elements-match }
```

La règle renvoie *vrai* si et seulement si au moins une valeur de l'attribut contient les éléments qui comprennent la valeur présentée, et si les valeurs des éléments présentés correspondent à celles des éléments homologues dans la valeur enregistrée. La comparaison des éléments est réalisée conformément aux règles définies pour la règle OR-address-match.

NOTE 1 – Il faut faire attention lors de la construction d'un filtre complexe. Lorsqu'un filtre est composé de deux éléments de filtre, chacun contenant un sous-ensemble d'éléments d'adresse OR et l'opérateur «et» **and** (par exemple «C=xx **and** O=zz»), chaque élément de filtre pourrait correspondre à différentes valeurs d'attribut. Par conséquent, une entrée pourrait être sélectionnée même si aucune valeur d'attribut ne contient les deux éléments en même temps; cela se produit si une valeur d'attribut contient C=xx et une autre O=zz. Cela est évité si les deux éléments sont présentés dans la même règle OR-address-elements-match.

Pour les composantes complexes (nom personnel, noms d'unités organisationnelles, attributs définis par domaine, et la forme étendue de l'adresse réseau), seuls les éléments présentés sont pris en considération.

NOTE 2 – Cela permet de procéder à un criblage sur une partie d'une composante complexe, par exemple «PersonalName.surname=Maruba».

12.4.3 Crible d'éléments de sous-chaînes d'adresse OR

La règle **OR-address-substring-elements-match** (crible d'éléments de sous-chaînes d'adresse OR) détermine si une valeur présentée est un sous-ensemble des éléments présents dans une certaine valeur d'un attribut de type OR-address (adresse OR), où chaque valeur de chaîne présentée est une sous-chaîne de la valeur enregistrée homologue.

```
oRAddressSubstringElementsMatch MATCHING-RULE ::= {
  SYNTAX      ORAddress
  ID          id-mr-oraddress-substring-elements-match }
```

Cette règle est identique à la règle OR-address-elements-match sauf que pour les éléments qui sont comparés par la règle MS-string-match, c'est la règle MS-single-substring-match qui est appliquée.

NOTE – Par exemple, la règle renvoie *vrai* lorsque la valeur présentée «OrganizationName=rc», «PersonalName.surname=arthur» est comparée à un attribut qui inclut la valeur «OrganizationName=RCC», «PersonalName.surname=McArthur».

12.4.4 Crible de nom OR

La règle **OR-name-match** (crible de nom OR) compare une valeur présentée aux valeurs d'attribut de type OR-name (nom OR) pour déterminer si elles sont égales.

```
oRNameMatch MATCHING-RULE ::= {
  SYNTAX      ORName
  ID          id-mr-orname-match }
```

La règle renvoie *vrai* si et seulement si la valeur présentée et au moins une valeur de l'attribut contiennent un nom OR qui vérifie les règles suivantes:

- a) si le nom OR présenté ne contient qu'une adresse OR, la règle est vérifiée si la valeur présentée correspond à une adresse OR dans la valeur enregistrée conformément à la règle OR-address-match;
- b) si le nom OR présenté ne contient qu'un nom d'annuaire, la règle est vérifiée si la valeur présentée correspond à un nom d'annuaire dans la valeur enregistrée conformément à la règle Distinguished Name Match;
- c) si le nom OR présenté contient à la fois une adresse OR et un nom d'annuaire, la règle est vérifiée si chacune des règles décrites ci-dessus est vérifiée.

Sinon la comparaison échoue.

NOTE – Lorsque le nom d'annuaire contient une valeur d'attribut dont le type est inconnu de la mémoire de messages, il ne sera pas toujours possible pour la MS d'exécuter une comparaison visant à vérifier la règle Distinguished Name Match.

12.4.5 Crible d'éléments de nom OR

La règle **OR-name-elements-match** (crible d'éléments de nom OR) détermine si une valeur présentée est un sous-ensemble des éléments présents dans une certaine valeur d'un attribut de type OR-name (nom OR).

```
oRNameElementsMatch MATCHING-RULE ::= {
  SYNTAX  ORName
  ID      id-mr-orname-elements-match }
```

La règle renvoie *vrai* si et seulement si au moins une valeur de l'attribut inclut les éléments qui comprennent la valeur présentée, et si les valeurs des éléments présentés correspondent à celles des éléments homologues dans la valeur enregistrée.

Cette règle est identique à la règle OR-name-match sauf que la règle OR-address-elements-match est utilisée au lieu de la règle OR-address-match.

NOTE – Au niveau local, une mémoire de messages peut élargir la définition de la règle Distinguished Name Match de sorte que seuls les noms distinctifs relatifs qui apparaissent dans la valeur présentée soient obligés de correspondre à leurs éléments homologues.

12.4.6 Crible d'éléments de sous-chaînes de nom OR

La règle **OR-name-substring-elements-match** (crible d'éléments de sous-chaînes de nom OR) détermine si une valeur présentée est un sous-ensemble des éléments présents dans une certaine valeur d'un attribut de type OR-name (nom OR), où chaque valeur de chaîne présentée est une sous-chaîne de la valeur enregistrée homologue.

```
oRNameSubstringElementsMatch MATCHING-RULE ::= {
  SYNTAX  ORName
  ID      id-mr-orname-substring-elements-match }
```

Cette règle est identique à la règle OR-name-elements-match sauf que pour les éléments qui sont comparés par la règle MS-string-match, c'est la règle MS-single-substring-match qui est appliquée.

NOTE – Au niveau local, une mémoire de messages peut élargir la définition de la règle Distinguished Name Match de sorte que seuls les noms distinctifs relatifs qui apparaissent dans la valeur présentée soient obligés de correspondre comme sous-chaînes de la valeur enregistrée.

12.4.7 Crible d'éléments individuels de nom OR

La règle **OR-name-single-element-match** (crible d'éléments individuels de nom OR) compare une chaîne présentée à un certain élément d'adresse OR présent dans la composante d'adresse OR d'une certaine valeur d'un attribut de type OR-name (nom OR) pour déterminer s'ils sont égaux.

```
oRNameSingleElementMatch MATCHING-RULE ::= {
  SYNTAX  MSString {ub-msstring-match}
  ID      id-mr-orname-single-element-match }
```

La règle renvoie *vrai* si et seulement si le nom OR enregistré contient une composante d'adresse OR dans laquelle au moins un élément d'adresse OR correspond à la valeur présentée conformément à la règle MS-string-match. Le type de terminal et la forme étendue des éléments de l'adresse réseau ne sont pas pris en considération lors de l'évaluation de la règle OR-name-single-element-match.

12.4.8 Crible de réacheminement ou de développement de liste de distribution

La règle **Redirection-or-DL-expansion-match** (crible de réacheminement ou de développement de liste de distribution) compare une valeur présentée à la composante OR-address-and-optional-directory-name (adresse OR et nom d'annuaire optionnel) des valeurs d'attribut de type Redirection-history (chronologie du réacheminement) ou DL-expansion-history (chronologie du développement de la liste de distribution) pour déterminer si elles sont égales.

```
redirectionOrDLExpansionMatch MATCHING-RULE ::= {
  SYNTAX  ORAddressAndOptionalDirectoryName
  ID      id-mr-redirection-or-dl-expansion-match }
```

La règle renvoie *vrai* si et seulement si la valeur présentée et au moins une valeur de l'attribut contiennent une composante OR-address-and-optional-directory-name qui vérifie la règle OR-name-match.

12.4.9 Crible d'éléments du réacheminement ou du développement de liste de distribution

La règle **Redirection-or-DL-expansion-elements-match** (crible d'éléments du réacheminement ou du développement de liste de distribution) détermine si une valeur présentée est un sous-ensemble des éléments présents dans la composante OR-address-and-optional-directory-name (adresse OR et nom d'annuaire optionnel) d'une certaine valeur d'un attribut de type Redirection-history (chronologie du réacheminement) ou DL-expansion-history (chronologie du développement de la liste de distribution).

```
redirectionOrDLExpansionElementsMatch MATCHING-RULE ::= {
  SYNTAX  ORAddressAndOptionalDirectoryName
  ID      id-mr-redirection-or-dl-expansion-elements-match }
```

La règle renvoie *vrai* si et seulement si la valeur présentée et au moins une valeur de l'attribut contiennent une composante OR-address-and-optional-directory-name qui vérifie la règle OR-name-elements-match.

12.4.10 Crible d'éléments de sous-chaînes du réacheminement ou du développement de liste de distribution

La règle **Redirection-or-DL-expansion-substring-elements-match** (crible d'éléments de sous-chaînes du réacheminement ou du développement de liste de distribution) détermine si une valeur présentée d'éléments de sous-chaînes est un sous-ensemble des éléments présents dans la composante OR-address-and-optional-directory-name (adresse OR et nom d'annuaire optionnel) d'une certaine valeur d'un attribut de type Redirection-history (chronologie du réacheminement) ou DL-expansion-history (chronologie du développement de la liste de distribution).

```
redirectionOrDLExpansionSubstringElementsMatch MATCHING-RULE ::= {
  SYNTAX  ORAddressAndOptionalDirectoryName
  ID      id-mr-redirection-or-dl-expansion-substring-elements-match }
```

Cette règle est identique à la règle Redirection-or-DL-expansion-elements-match sauf que pour les éléments qui sont comparés par la règle MS-string-match, c'est la règle MS-single-substring-match qui est appliquée.

12.4.11 Crible de motifs de réacheminement

La règle **Redirection-reason-match** (crible de motifs de réacheminement) compare une valeur présentée à la composante Redirection-reason (motif de réacheminement) des valeurs d'attribut de type Redirection-history (chronologie du réacheminement) pour déterminer si elles sont égales.

```
redirectionReasonMatch MATCHING-RULE ::= {
  SYNTAX  RedirectionReason
  ID      id-mr-redirection-reason-match }
```

La règle renvoie *vrai* si et seulement si la valeur présentée correspond à la composante redirection-reason (motif de réacheminement) d'au moins une valeur de l'attribut conformément à la règle Integer Match.

12.4.12 Crible d'identificateur MTS

La règle **MTS-identifiant-match** (crible d'identificateur MTS) compare une valeur présentée aux valeurs d'attribut de type MTS-identifiant (identificateur MTS) pour déterminer si elles sont égales.

```
mTSIdentifiantMatch MATCHING-RULE ::= {
  SYNTAX  MTSIdentifiant
  ID      id-mr-mts-identifiant-match }
```

Chaque élément de type MTS-identifiant (nom de pays, nom du domaine d'administration, identificateur du domaine privé et identificateur local) est comparé par la règle MS-string-match. La règle renvoie *vrai* si et seulement si au moins une valeur de l'attribut contient des éléments qui correspondent à ceux de la valeur présentée. La règle d'équivalence entre les formes X.121 et ISO 3166 de l'élément country-name (nom de pays) peut être appliquée (voir la Rec. UIT-T X.402 | ISO/CEI 10021-2).

12.4.13 Crible de corrélateur de contenu

La règle **Content-correlator-match** (crible de corrélateur de contenu) compare la valeur présentée aux valeurs d'attribut de type content-correlator (corrélateur de contenu) pour déterminer si elles sont égales.

```
contentCorrelatorMatch MATCHING-RULE ::= {
  SYNTAX  ContentCorrelator
  ID      id-mr-content-correlator-match }
```

Si les valeurs présentée et enregistrée sont codées comme des chaînes d'octets Octet Strings, la règle doit vérifier la règle Octet String Match. Si les deux valeurs sont codées comme des chaînes de caractères, la règle doit vérifier la règle MS-string-case-sensitive-match. Sinon la comparaison échoue.

12.4.14 Crible d'identificateur de contenu

La règle **Content-identifiant-match** (crible d'identificateur de contenu) compare la valeur présentée aux valeurs d'attribut de type content-identifiant (identificateur de contenu) pour déterminer si elles sont égales.

```
contentIdentifiantMatch MATCHING-RULE ::= {
  SYNTAX  ContentIdentifiant
  ID      id-mr-content-identifiant-match }
```

La règle renvoie *vrai* si et seulement si la valeur présentée correspond à au moins une valeur de l'attribut conformément à la règle MS-string-case-sensitive-match.

12.5 Prise en charge des règles de criblage

La prise en charge des règles de criblage générales suivantes est obligatoire: Integer Match, Integer Ordering Match, Object Identifier Match, UTC Time Match et UTC Ordering Match. Une mémoire de messages qui déclare prendre en charge une règle de criblage doit prendre en charge son utilisation dans un filtre pour tout type d'attribut qu'elle déclare également prendre en charge, quand le type d'attribut inclut cette règle de criblage dans la définition de son attribut; voir 11.1.1.

Le présent crible est pris en charge pour tous les types d'attribut pris en charge.

12.6 Ensemble d'objets informationnels (Matching-rule-table)

L'ensemble d'objets informationnels **Matching-rule-table** (tableau des règles de criblage) est utilisé comme un ensemble de contraintes dans la présente Définition de service lorsque des champs liés à la classe d'objets informationnels MATCHING-RULE (règle de criblage) sont référencés dans la syntaxe abstraite MS. Il comprend deux ensembles d'objets:

```
MatchingRuleTable MATCHING-RULE ::= {
  GeneralMatchingRules | ContentSpecificMatchingRules }
```

L'ensemble d'objets **General-matching-rules** (règles de criblage générales) contient les attributs généraux définis dans la présente Définition de service. Il est défini de la façon suivante:

```
GeneralMatchingRules MATCHING-RULE ::= {
  bitStringMatch | booleanMatch | contentIdentifiantMatch | integerMatch | integerOrderingMatch |
  mSStringCaseSensitiveMatch | objectIdentifiantMatch | oRNameMatch | uTCTimeMatch |
  uTCTimeOrderingMatch,
  ... -- extensions de 1994 --,
  contentCorrelatorMatch | mSSingleSubstringMatch | mSStringCaseSensitiveMatch |
  mSStringListElementsMatch | mSStringListMatch | mSStringMatch | mSStringOrderingMatch |
  mSSingleSubstringListElementsMatch | mSSingleSubstringListMatch | mSSubstringsMatch |
  mTIdentifierMatch | oRAddressElementsMatch | oRAddressMatch | oRAddressSubstringElementsMatch |
  oRNameElementsMatch | oRNameMatch | oRNameSingleElementMatch | oRNameSubstringElementsMatch |
  redirectionOrDLExpansionElementsMatch | redirectionOrDLExpansionMatch |
  redirectionOrDLExpansionSubstringElementsMatch | redirectionReasonMatch | valueCountMatch }
```

L'ensemble d'objets **Content-specific-matching-rules** (règles de criblage propres au contenu) est un ensemble vide et extensible. Lorsqu'une mémoire de messages offre la prise en charge d'un type de contenu donné, les règles de criblage associées à ce type de contenu doivent être considérées comme des objets appartenant à l'ensemble d'objets Content-specific-matching-rules.

```
ContentSpecificMatchingRules MATCHING-RULE ::= { ... }
```

13 Actions automatiques générales

Les **actions automatiques générales** sont applicables aux entrées de tous les types de contenu. D'autres actions automatiques, propres à un type de contenu sont définies dans la Spécification appropriée.

Les **actions automatiques** sont présentées au 6.5. La consignation et la déconsignation d'actions automatiques au moyen de l'opération abstraite Register-MS (consignation MS) sont décrites au 8.2.5. En variante, les informations de consignation peuvent être acheminées dans la mémoire de messages par abonnement. Cependant, certains types d'actions automatiques peuvent nécessiter que la mémoire de messages prenne en charge la consignation au moyen de l'opération abstraite Register-MS.

Les types d'actions automatiques générales suivants sont définis:

- a) Auto-alert (avertissement automatique);
- b) Auto-modify (modification automatique);
- c) Auto-correlate-reports (rapports de corrélation automatique);
- d) Auto-delete (suppression automatique).

NOTE – L'action automatique Auto-forward (retransmission automatique) définie dans les précédentes éditions de cette Spécification est maintenant définie comme l'action automatique IPM auto-forward (retransmission automatique de messages de personne à personne) dans la Rec. UIT-T X.420 | ISO/CEI 10021-7.

Chaque type d'action automatique générale est défini comme une instance de la classe d'objets informationnels AUTO-ACTION (action automatique) (voir 6.5.1). L'ensemble d'objets informationnels **Auto-action-table** (tableau d'actions automatiques) est utilisé comme un ensemble de contraintes dans la présente Définition de service quand des champs liés à la classe d'objets informationnels AUTO-ACTION sont référencés dans la syntaxe abstraite MS. Il comprend deux ensembles d'objets:

```
AutoActionTable AUTO-ACTION ::= {
  GeneralAutoActions | ContentSpecificAutoActions }
```

L'ensemble d'objets **General-auto-actions** (actions automatiques générales) contient les actions automatiques générales définies dans la présente Définition de service. Il est défini de la façon suivante:

```
GeneralAutoActions AUTO-ACTION ::= {
  auto-alert,
  ... -- extensions de 1994 --,
  auto-modify | auto-correlate-reports | auto-delete }
```

L'ensemble d'objets **Content-specific-auto-actions** (actions automatiques propres au contenu) est un ensemble vide et extensible. Quand une mémoire de messages offre la prise en charge d'un type de contenu donné, les actions automatiques associées à ce type de contenu doivent être considérées comme des objets appartenant à l'ensemble d'objets Content-specific-auto-actions.

```
ContentSpecificAutoActions AUTO-ACTION ::= { ... }
```

Chaque erreur d'action automatique qui sera créée par les types d'action automatique générale est définie comme une instance des classes d'objets informationnels AUTO-ACTION-ERROR (erreur d'action automatique) ou ABSTRACT-ERROR (erreur abstraite) (les deux sont identiques); voir 6.5.3. L'ensemble d'objets informationnels **Auto-action-error-table** (tableau des erreurs d'action automatique) est défini ci-dessous. Il comprend les ensembles d'objets informationnels General-auto-action-errors (erreurs d'action automatique générale) et Content-specific-auto-action-errors (erreurs d'action automatique propre au contenu):

```
AutoActionErrorTable AUTO-ACTION-ERROR ::= {
  GeneralAutoActionErrors | ContentSpecificAutoActionErrors }
```

```
GeneralAutoActionErrors AUTO-ACTION-ERROR ::= {
  autoAlertError | modifyError | serviceError | securityError | messageGroupError,
  ... -- Réserve pour des extensions futures -- }
```

```
ContentSpecificAutoActionErrors AUTO-ACTION-ERROR ::= { ... }
```

Le fonctionnement des actions automatiques peut être affecté par l'application d'une politique de sécurité.

Sauf déclaration contraire explicite, l'exécution d'une action automatique n'a aucun effet sur la valeur de l'attribut général retrieval-status (état d'extraction) d'une entrée. Aucune des actions automatiques définies dans la présente Définition de service n'affecte la valeur de l'attribut général retrieval-status.

13.1 Avertissement automatique Auto-alert

L'action automatique **Auto-alert** (avertissement automatique) permet à l'utilisateur MS de commander à la mémoire de messages d'avertir automatiquement l'utilisateur lorsque la remise de messages ou de rapports sélectionnés entraîne la création d'une entrée dans la classe d'entrées Delivery (remise).

L'avertissement peut être acheminé vers l'utilisateur MS à l'aide de l'opération abstraite Alert (avertissement) (voir 8.2.6). En variante, (ou en plus), l'avertissement peut être acheminé vers l'entité qui utilise les services de l'utilisateur MS (par exemple une personne) à l'aide d'un autre mécanisme d'avertissement, par exemple un appel de personne, un voyant lumineux sur un téléphone ou un autre équipement terminal. Dans ce cas, le dispositif qui reçoit l'avertissement est appelé **alert-destination** (destination de l'avertissement) et il est identifié par une adresse d'avertissement **alert-address**.

Le type d'action automatique **Auto-alert** permet de consigner avec la mémoire de messages un ou plusieurs paramètres de consignation d'avertissement automatique auto-alert-registration-parameter, chacun étant identifié par son identificateur de consignation d'avertissement automatique. Les informations de consignation peuvent être acheminées dans la mémoire de messages en utilisant l'opération abstraite Register-MS (consignation MS) ou par abonnement. Chaque paramètre auto-alert-registration parameter spécifie des critères permettant de déterminer s'il s'applique à tel ou tel message ou rapport remis. Il peut également spécifier une ou plusieurs destinations d'avertissement et les attributs du message ou du rapport remis qui doivent être acheminés dans l'opération abstraite Alert (avertissement).

```

auto-alert AUTO-ACTION ::= {
  REGISTRATION PARAMETER IS AutoAlertRegistrationParameter
  ERRORS {auto-alert-error}
  IDENTIFIED BY id-act-auto-alert }

AutoAlertRegistrationParameter ::= SET {
  filter [0] Filter OPTIONAL,
  alert-destinations [1] SEQUENCE SIZE (1..ub-alert-addresses) OF
    AlertDestination OPTIONAL,
  requested-attributes [2] EntryInformationSelection OPTIONAL,
  -- extension de 1994 --
  suppress-alert-destinations [3] BOOLEAN DEFAULT TRUE }

```

Les composantes du paramètre de consignation d'avertissement automatique **auto-alert-registration parameter** ont la signification suivante:

- a) **filter** (filtre) (optionnel): cette composante spécifie un ensemble de critères auxquels une nouvelle entrée représentant un rapport ou un message remis doit être conforme avant que la mémoire de messages exécute cette action automatique Auto-alert (avertissement automatique) consignée. En l'absence de cette composante, cette action automatique Auto-alert sera exécutée pour tous les nouveaux messages et rapports remis;
- b) **alert-destinations** (destinations d'avertissement) (optionnel): cette composante identifie une ou plusieurs destinations d'avertissement, chacune étant identifiée par une adresse d'avertissement, ainsi que toute information nécessaire pour commander l'avertissement et décrire les informations à acheminer vers la destination de l'avertissement. La mémoire de messages doit essayer d'avertir les destinations d'avertissement sous réserve des indications spécifiées au d).

```

AlertDestination ::= SEQUENCE {
  alert-address EXTERNAL,
  alert-qualifier OCTET STRING OPTIONAL }

```

L'adresse d'avertissement **alert-address** identifie le type d'avertissement à effectuer (par exemple le système d'appel de personne à utiliser) et l'adresse d'une entité qu'il est possible d'atteindre par un avertissement de ce type. Le qualificatif d'avertissement **alert-qualifier** spécifie toute information supplémentaire requise pour commander l'avertissement, et d'autres informations à acheminer vers la destination de l'avertissement. En l'absence de cette composante, aucune information de commande n'est spécifiée, et aucune information supplémentaire ne sera acheminée vers la destination de l'avertissement (sauf si les informations requises ont été consignées par abonnement).

Certains types d'avertissements peuvent être normalisés au niveau international. D'autres peuvent être définis par les autorités administratives nationales ou par des organisations privées. Il s'ensuit que plusieurs autorités distinctes seront responsables de l'affectation des identificateurs pour chaque type d'avertissement et devront s'assurer que chaque type est unique. Chaque type d'avertissement est identifié par un identificateur d'objet qui est affecté lorsque le type d'avertissement est défini. L'identificateur d'objet figure dans la composante alert-address (adresse d'avertissement). Le type de données ASN.1 de l'adresse d'avertissement et d'autres règles permettant de traiter le qualificatif d'avertissement alert-qualifier sont définis lorsque le type d'avertissement est défini;

- c) **requested-attributes** (attributs demandés) (optionnel): cette composante indique les informations du message ou du rapport remis à inclure dans l'argument de l'opération abstraite Alert (avertissement); voir 8.1.4.

L'opération abstraite Alert doit être invoquée si cette composante est présente ou si la composante alert-destinations (destinations d'avertissement) est absente;

- d) **suppress-alert-destinations** (suppression des destinations d'avertissement) (optionnel): cette composante n'a un sens que si les deux composantes alert-destinations et requested-attributes sont présentes; sinon il n'en est pas tenu compte. Si cette composante est *vraie*, et que l'invocation de l'opération abstraite Alert (avertissement) a abouti, il ne faut pas essayer d'acheminer l'avertissement vers les destinations d'avertissement.

Lorsqu'un message ou un rapport est remis, les actions automatiques Auto-alert consignées sont exécutées chacune à leur tour, dans l'ordre de leurs identificateurs de consignation, jusqu'à ce que l'une d'elles ait abouti. Cela se produit lorsque le message ou le rapport remis est conforme aux critères de sélection de l'action Auto-alert consignée, et que l'utilisateur MS ou l'une des destinations d'avertissement est averti comme cela est décrit ci-dessus. Si la composante requested-attributes (attributs demandés) est présente ou que la composante alert-destinations (destinations d'avertissement) est absente, l'opération abstraite Alert doit être invoquée. Cela n'est possible que si une association abstraite existe déjà entre la mémoire de messages et l'utilisateur MS. Si la composante alert-destinations est présente [et que l'exécution de l'opération Alert n'a pas abouti, ou que la composante suppress-alert-destinations (suppression des destinations d'avertissement) est *fausse*], la mémoire de messages essaie de fournir l'avertissement aux destinations d'avertissement chacune à leur tour, jusqu'à ce qu'une tentative aboutisse. Si, après le traitement de toutes les consignations Auto-alert, l'entrée s'est révélée conforme aux critères de sélection d'au moins une consignation, mais si aucune des tentatives pour acheminer l'avertissement à une destination d'avertissement, ou pour invoquer l'opération abstraite Alert n'a abouti, la mémoire de messages doit positionner l'indication d'avertissement qui est signalée à l'utilisateur MS lors de l'établissement suivant d'une association abstraite; voir 7.1.2 d).

L'erreur d'action automatique suivante est créée si aucune des destinations d'avertissement ne peut être avertie, ou si une certaine condition d'erreur définie localement surgit:

```

auto-alert-error AUTO-ACTION-ERROR ::= {
  PARAMETER   SEQUENCE SIZE (1..ub-alert-addresses) OF AutoAlertErrorIndication
  CODE        global:id-aae-auto-alert-error }

AutoAlertErrorIndication ::= SET {
  failing-alert-destination   [0] AlertDestination OPTIONAL,
  supplementary-information   [1] GeneralString (SIZE (1..ub-supplementary-info-length)) OPTIONAL }

```

Le paramètre a la signification suivante:

- a) **failing-alert-destination** (destination d'avertissement où se trouve une erreur) (optionnel): identifie la destination d'avertissement, le cas échéant, associée à la condition d'erreur décrite dans la composante supplementary-information;
- b) **supplementary-information** (information complémentaire) (optionnel): description de la condition d'erreur.

NOTE – Au niveau local, la mémoire de messages peut créer une erreur d'action automatique concernant une destination d'avertissement même si une autre destination d'avertissement a bien été avertie, par exemple lorsqu'une destination d'avertissement est consignée avec une adresse erronée qui ne peut être détectée que quand on essaie d'avertir cette destination.

L'exécution de l'action automatique Auto-alert (avertissement automatique) peut entraîner la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques).

13.2 Modification automatique (Auto-modify)

L'action automatique **Auto-modify** (modification automatique) permet à l'utilisateur MS d'ordonner à la mémoire de messages d'appliquer automatiquement des modifications aux attributs d'une entrée à condition que l'entrée soit conforme à des critères donnés. L'action automatique est exécutée chaque fois qu'une entrée des classes d'entrée Stored-message (message enregistré) ou Message-log (journal de messagerie) est créée, c'est-à-dire lorsqu'un message ou un rapport est remis, ou lorsqu'un brouillon de message est enregistré, ou lorsqu'un message ou un message d'essai est déposé et enregistré. L'action automatique Auto-modify s'applique également aux entrées créées à la suite de l'exécution d'actions automatiques propres au contenu.

Seuls certains types d'attributs généraux peuvent être modifiés au moyen de cette action automatique (voir 11.6). Les types d'attribut propres à un type de contenu particulier, et pouvant être modifiés, sont définis dans la Spécification appropriée. La définition d'un type d'attribut pouvant être modifié peut spécifier des règles supplémentaires pour l'exécution de l'action automatique Auto-modify (modification automatique).

Le type d'action automatique Auto-modify permet de consigner avec la mémoire de messages un ou plusieurs paramètres de consignation de modification automatique auto-modify-registration-parameter, chacun étant identifié par son identificateur de consignation. Chaque paramètre auto-modify-registration-parameter identifie la classe d'entrées appelée par l'action automatique, et spécifie des critères qui déterminent si l'action automatique doit s'appliquer à une entrée particulière, auquel cas l'entrée est soumise à l'ensemble correspondant de modifications. Pour chaque action automatique Auto-modify consignée pour laquelle l'entrée est conforme aux critères de sélection, l'ensemble correspondant de modifications est appliqué.

```

auto-modify AUTO-ACTION ::= {
  REGISTRATION PARAMETER IS AutoModifyRegistrationParameter
  ERRORS {securityError | serviceError | modifyError |
            messageGroupError}
  IDENTIFIED BY id-act-auto-modify }

AutoModifyRegistrationParameter ::= SET {
  entry-class [0] EntryClass DEFAULT delivery,
  filter [1] Filter OPTIONAL,
  modifications [2] SEQUENCE SIZE (1..ub-modifications) OF EntryModification }

```

Les composantes du paramètre de consignation de modification automatique **auto-modify-registration-parameter** ont la signification suivante:

- a) **entry-class** (classe d'entrée) (optionnel): cette composante spécifie la classe d'entrées appelée par l'action automatique;
- b) **filter** (filtre) (optionnel): cette composante spécifie un ensemble de critères auquel une nouvelle entrée doit être conforme avant que la mémoire de messages n'exécute cette action automatique Auto-modify (modification automatique) consignée. En l'absence de cette composante, toutes les entrées créées dans la classe d'entrées spécifiée peuvent être modifiées;
- c) **modifications** (obligatoire): cette composante définit une séquence de modifications, qui seront appliquées dans l'ordre spécifié dans la séquence (voir 8.2.7.1).

L'exécution de l'action automatique Auto-modify peut entraîner la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques).

La prise en charge de l'action automatique Auto-modify par une mémoire de messages, ou par un agent utilisateur (UA) accédant à une mémoire de messages, nécessite la prise en charge de la consignation du paramètre auto-modify-registration-parameter à l'aide de l'opération abstraite Register-MS (consignation MS). Les informations de consignation peuvent également être acheminées dans la mémoire de messages par abonnement.

13.3 Corrélation automatique des rapports (Auto-correlate-reports)

L'action automatique **Auto-correlate-reports** (corrélation automatique des rapports) permet à l'utilisateur MS de commander à la mémoire de messages de corréler automatiquement les entrées delivered-report (rapport remis) de la classe d'entrées Delivery (remise) avec les messages et les messages d'essai déposés auxquels ils se rapportent. L'action automatique enregistre également chaque invocation de l'opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS) qui a abouti. Le type d'action automatique n'est disponible que sur abonnement, et non par consignation à l'aide de l'opération abstraite Register-MS (consignation MS).

```

auto-correlate-reports AUTO-ACTION ::= {
  IDENTIFIED BY id-act-auto-correlate-reports }

```

La mémoire de messages prend en charge l'action automatique Auto-correlate-reports à l'aide des attributs généraux AC-correlated-report-list, AC-report-summary, AC-uncorrelated-report-list et deferred-delivery-cancellation-time; voir 11.2.1, 11.2.2, 11.2.3 et 11.2.20.

Aucune entrée ne doit être créée dans la classe d'entrées Auto-action-log (journal d'actions automatiques) comme résultat de l'exécution de l'action automatique Auto-correlate-reports.

13.4 Suppression automatique (Auto-delete)

L'action automatique **Auto-delete** (suppression automatique) permet à l'utilisateur MS d'ordonner à la mémoire de messages de supprimer automatiquement une entrée (et toutes les entrées filles qui lui sont associées) au terme d'un délai prédéterminé après sa création. La mémoire de messages doit soumettre à la suppression automatique chaque entrée dans

la classe d'entrées Stored-message (message enregistré) qui possède un attribut général storage-time (heure d'enregistrement) dont la valeur est inférieure aux date et heure courantes. La suppression automatique ne doit pas être effectuée sur une association abstraite existante entre la mémoire de messages et l'utilisateur MS. Toutes les restrictions spécifiées pour l'opération abstraite Delete (suppression) s'appliquent également à l'action automatique Auto-delete. Par conséquent, l'action Auto-delete ne doit pas s'appliquer aux entrées filles ni aux entrées de la classe Delivery (remise) dont la valeur de l'attribut retrieval-status (état d'extraction) est *nouveau* (voir 8.2.4).

```

auto-delete AUTO-ACTION ::= {
  ERRORS           {securityError}
  IDENTIFIED BY   id-act-auto-delete }

```

Aucun paramètre de consignation n'est défini pour l'action automatique Auto-delete (suppression automatique).

NOTES

1 Cette action automatique diffère des autres actions automatiques définies dans la présente Définition de service dans le sens où elle devient active à l'expiration d'une temporisation et non à la création d'une entrée. La période pendant laquelle la suppression doit se produire après que l'heure d'enregistrement est atteinte n'est pas prescrite. En pratique, les applications peuvent choisir d'exécuter cette action à intervalles réguliers, par exemple toutes les heures, tous les jours ou toutes les semaines.

2 Une entrée nouvellement créée n'est soumise à l'exécution de l'action Auto-delete qu'une fois que toutes les autres actions automatiques consignées ont été exécutées.

3 Une entrée qui contient l'attribut général storage-period (période d'enregistrement) contient également l'attribut général storage-time (heure d'enregistrement). Les deux attributs indiquent, de deux façons différentes, la date et l'heure après laquelle le message doit être supprimé, c'est-à-dire que la combinaison des attributs storage-period et creation-time (heure de création) est équivalente à l'attribut storage-time; voir 11.2.71.

4 Le type d'attribut marked-for-deletion (marqué pour être supprimé) n'est pas pris en considération lors de l'exécution de l'action automatique Auto-delete (suppression automatique).

La prise en charge de l'action automatique Auto-delete (suppression automatique) par une mémoire de messages, ou par un agent utilisateur (UA) accédant à une mémoire de messages, nécessite la prise en charge de la consignation de l'action automatique avec la mémoire de messages à l'aide de l'opération abstraite Register-MS (consignation MS). L'action automatique Auto-delete est également disponible par abonnement.

L'exécution de l'action automatique Auto-delete peut entraîner la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques).

SECTION 4 – PROCÉDURES APPLICABLES À L'ENREGISTREMENT DE MESSAGES ET À LA RÉALISATION DES ACCÈS

14 Présentation générale

La présente section décrit les procédures applicables à la mémoire de messages et à la réalisation des accès. On trouve à l'article 15 une description de l'utilisation du service abstrait MTS. La fourniture du service abstrait MS est décrite à l'article 16. La réalisation des accès sous forme d'éléments de service d'application est décrite à l'article 17.

Le fonctionnement des opérations abstraites décrites aux articles 15 et 16 peut être soumis aux exigences d'une politique de sécurité (si une telle politique est en vigueur) qui s'applique au service abstrait MTS et au service abstrait MS.

15 Utilisation du service abstrait du système de transfert de messages

Le présent article spécifie comment la mémoire de messages utilise le service abstrait MTS qui est défini à l'article 8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Il définit l'utilisation par la mémoire de messages des services des accès MTS de remise, dépôt et administration.

15.1 Utilisation des services abstraits de l'accès de remise

Ce paragraphe décrit le fonctionnement des opérations abstraites Message-delivery (remise de message) et Report-delivery (remise de rapport), ainsi que l'invocation de l'opération abstraite Delivery-control (commande de remise). L'utilisation par la mémoire de messages des services abstraits de l'accès de remise suppose qu'il existe une association abstraite entre le fournisseur de l'accès de remise (le système MTS) et le client de l'accès de remise (la mémoire de messages). Ces opérations abstraites sont exécutées séquentiellement et il n'y a pas de traitement parallèle. Les cas d'erreur ne sont pas décrits.

15.1.1 Fonctionnement de l'opération abstraite Message-delivery (remise de message)

Lorsque le système MTS invoque l'opération abstraite Message-delivery (remise de message), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages crée une entrée (et éventuellement une ou plusieurs entrées filles) de type delivered-message (message remis) dans la classe d'entrées Delivery (remise) et positionne son état d'extraction à *nouveau*. Une entrée correspondante de même numéro de séquence est créée dans la classe d'entrées Delivery-log (journal de remise). Les types d'attribut obligatoires et optionnels pour les entrées delivered-message des classes Delivery et Delivery-log sont indiqués dans le Tableau 2;
- b) la mémoire de messages renvoie un résultat de remise de message au système MTS pour indiquer que la remise du message a abouti. Ce résultat doit contenir une information de preuve de remise si le message remis contient un argument de demande de preuve de remise. Cette preuve de remise peut être évaluée à l'aide de la clé secrète du sujet MS; voir 8.5.7 et 8.3.1.1.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- c) la mémoire de messages détermine si des actions automatiques doivent être exécutées. En plus des types d'actions automatiques générales définis dans la présente Définition de service, des types d'actions automatiques propres au contenu peuvent être définis pour des types de contenu particuliers dans les Spécifications appropriées. A l'exception des actions automatiques propres au contenu dont l'ordre d'exécution est spécifié dans la Spécification appropriée, les actions automatiques sont exécutées dans l'ordre indiqué ci-dessous:
 - 1) si des actions automatiques Auto-modify (modification automatique) ont été consignées, la nouvelle entrée est confrontée à la classe d'entrées et aux critères de sélection spécifiés dans chaque paramètre de consignation de l'action Auto-modify. Si l'entrée est conforme à ces critères, la séquence des modifications de l'entrée est appliquée. Cela est répété pour chaque action automatique Auto-modify consignée;
 - 2) si des actions automatiques Auto-alert (avertissement automatique) ont été consignées, la nouvelle entrée est confrontée aux critères de sélection spécifiés dans chaque paramètre de consignation de l'action Auto-alert. Si l'entrée est conforme aux critères de sélection de l'un quelconque de ces paramètres, la mémoire de messages essaiera d'invoquer l'opération abstraite Alert ou de fournir l'avertissement à une destination d'avertissement (ou d'effectuer ces deux actions) comme cela est défini au 13.1.

NOTE – Si le message remis est supprimé comme résultat d'une action automatique propre au contenu, l'action Auto-alert n'est pas exécutée.

Certaines actions automatiques, lorsqu'elles sont exécutées, provoquent la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques). Si l'une de ces actions automatiques entraîne une erreur d'action automatique, la mémoire de messages attachera un attribut auto-action-error (erreur d'action automatique) indiquant la nature de l'erreur à l'entrée de la classe Auto-action-log et positionnera le paramètre auto-action-error-indication (indication d'erreur d'action automatique), qui sera signalé à l'utilisateur MS lors de l'établissement de l'association abstraite suivante. La mémoire de messages reprend alors l'exécution des actions automatiques consignées;

- d) après l'exécution des actions décrites ci-dessus, la nouvelle entrée est rendue accessible à l'utilisateur MS à travers l'accès d'extraction.

15.1.2 Fonctionnement de l'opération abstraite Report-delivery (remise de rapport)

Lorsque le système MTS invoque l'opération abstraite Report-delivery (remise de rapport), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages crée une entrée de type delivered-report (rapport remis) dans la classe d'entrées Delivery (remise) et positionne son état d'extraction à *nouveau*. Une entrée correspondante de même numéro de séquence est créée dans la classe d'entrées Delivery-log (journal de remise). Les types d'attribut obligatoires et optionnels pour les entrées delivered-report des classes Delivery et Delivery-log sont indiqués dans le Tableau 2;

- b) si le rapport remis contient le message déposé au départ dans son champ returned-content (contenu renvoyé), la mémoire de messages créera une entrée fille contenant l'entrée returned-content dans la classe d'entrées Delivery (remise); aucune entrée n'est créée dans la classe d'entrées Delivery-log (journal de messagerie);
- c) la mémoire de messages renvoie un résultat de remise de rapport au système MTS pour indiquer que la remise a abouti. Ce résultat n'a pas de paramètre; voir 8.3.1.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021 4;
- d) la mémoire de messages détermine si des actions automatiques doivent être exécutées. Les actions automatiques sont exécutées dans l'ordre indiqué ci-dessous:
 - 1) si l'action automatique Auto-correlate-reports (rapports de corrélation automatique) est en cours d'exécution, la mémoire de messages essaie alors de corréler le rapport remis avec l'entrée de la classe Submission (dépôt) ou Submission-log (journal de dépôt) qui est le sujet du rapport. Si une entrée correspondant au message sujet du rapport est trouvée dans la classe d'entrées Submission ou Submission-log, la mémoire de messages mettra à jour les attributs de corrélation automatique indiqués au 13.3;
 - 2) si des actions automatiques Auto-modify (modification automatique) ont été consignées, la nouvelle entrée est confrontée à la classe d'entrées et aux critères de sélection spécifiés dans chaque paramètre de consignation de l'action Auto-modify. Si l'entrée est conforme à ces critères, la séquence des modifications de l'entrée est appliquée. Cela est répété pour chaque action automatique Auto-modify consignée;
 - 3) si des actions automatiques Auto-alert (avertissement automatique) ont été consignées, la nouvelle entrée est confrontée aux critères de sélection spécifiés dans chaque paramètre de consignation de l'action Auto-alert. Si l'entrée est conforme aux critères de sélection de l'un quelconque de ces paramètres, la mémoire de messages essaiera d'invoquer l'opération abstraite Alert ou de fournir l'avertissement à une destination d'avertissement (ou d'effectuer ces deux actions) comme cela est défini au 13.1.

Certaines actions automatiques, lorsqu'elles sont exécutées, provoquent la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques). Si l'une de ces actions automatiques entraîne une erreur d'action automatique, la mémoire de messages attachera un attribut auto-action-error (erreur d'action automatique) indiquant la nature de l'erreur à l'entrée de la classe Auto-action-log et positionnera le paramètre auto-action-error-indication (indication d'erreur d'action automatique), qui sera signalé à l'utilisateur MS lors de l'établissement de l'association abstraite suivante. La mémoire de messages reprend alors l'exécution des actions automatiques consignées;

- e) après l'exécution des actions décrites ci-dessus, la nouvelle entrée est rendue accessible à l'utilisateur MS à travers l'accès d'extraction.

15.1.3 Invocation de l'opération abstraite Delivery-control (commande de remise)

Si la mémoire de messages nécessite une restriction temporaire sur les opérations abstraites de l'accès de remise que le système MTS peut invoquer ou si elle nécessite une modification des restrictions existantes, elle effectue les opérations suivantes:

- a) la mémoire de messages invoque l'opération abstraite Delivery-control (commande de remise) contenant les paramètres à modifier; voir 8.3.1.3 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) le système MTS renvoie un résultat pour indiquer que les commandes spécifiées sont maintenant en vigueur; sinon une erreur est renvoyée. Le résultat indique les opérations abstraites que le système MTS invoquerait, ou les types de message qu'il remettrait en l'absence des commandes prédominantes; voir 8.3.1.3.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- c) lorsque la mémoire de messages est en mesure d'accepter des messages ou des rapports en attente de remise, elle peut invoquer l'opération abstraite Delivery-control (commande de remise) pour annuler les restrictions. Les effets d'une opération abstraite Delivery-control sont annulés lorsqu'une nouvelle opération abstraite Delivery-control modifie les restrictions ou lorsque l'association abstraite entre la mémoire de messages et le système MTS est libérée.

15.1.4 Règles de création applicables aux attributs généraux

Les attributs doivent être créés lorsqu'un message ou un rapport est remis ou peuvent être créés lorsqu'une action automatique est exécutée.

Les entrées des classes d'entrées Delivery (remise) et Delivery-log (journal de remise) créées comme résultat des opérations Message-delivery (remise de message) et Report-delivery (remise de rapport) doivent contenir les attributs

obligatoires indiqués dans le Tableau 2, et les attributs optionnels qui sont pris en charge par la réalisation de la mémoire de messages et auxquels l'utilisateur MS s'est abonné. Les attributs créés constituent une nouvelle entrée, ou dans certains cas une entrée mère et une ou plusieurs entrées filles, voir 6.3.4. Chaque attribut d'une entrée dans la classe d'entrées Delivery-log est identique à l'attribut de même nom dans l'entrée correspondante de la classe d'entrées Delivery [à l'exception de l'attribut deletion-time (heure de suppression) qui ne peut être présent que dans les entrées de la classe Delivery-log]. Les types suivants d'attributs généraux doivent être créés:

- a) attributs généraux créés par la mémoire de messages elle-même (par exemple, numéro de séquence);
- b) attributs généraux créés à partir des composantes de l'enveloppe de remise de message et de l'enveloppe de remise de rapport. Pour les composantes qui ne sont pas présentes mais pour lesquelles des valeurs par défaut sont définies, un attribut général contenant la valeur par défaut est créé.

On trouvera au Tableau 4 et au 11.4 les règles de création des attributs généraux. Les règles de création des attributs propres au contenu sont définies dans les Spécifications appropriées (par exemple, les attributs propres au système de messagerie de personne à personne sont décrits au 19.7 de la Rec. UIT-T X.420 | ISO/CEI 10021-7).

15.2 Utilisation des services abstraits de l'accès de dépôt

Ce paragraphe décrit l'invocation des opérations abstraites Message-submission (dépôt de message), Probe-submission (dépôt de message d'essai) et Cancel-deferred-delivery (annulation de remise différée) ainsi que l'utilisation de l'opération abstraite Submission-control (commande de dépôt). L'utilisation par la mémoire de messages des services abstraits de l'accès de dépôt suppose qu'une association abstraite existe entre le fournisseur de l'accès de dépôt (le système MTS) et le client de l'accès de dépôt (la mémoire de messages). Les opérations abstraites sont exécutées séquentiellement et aucun traitement parallèle n'a lieu. Les cas d'erreur ne sont pas décrits.

15.2.1 Invocation de l'opération abstraite Message-submission (dépôt de message)

L'invocation de l'opération abstraite Message-submission (dépôt de message) résulte de l'exécution d'une action automatique par la mémoire de messages ou de l'invocation de l'opération abstraite MS-message-submission (dépôt de message MS) par l'utilisateur MS. La définition de l'exécution de l'opération MS-message-submission inclut une description de l'invocation de l'opération abstraite Message-submission; voir 16.2.1.

Si l'invocation de l'opération Message-submission (dépôt de message) est la conséquence de l'exécution d'une action automatique, les opérations supplémentaires suivantes sont effectuées:

- a) le paramètre submission-options (options de dépôt) de l'opération abstraite MS-message-submission (dépôt de message MS) est obtenu de la façon suivante. Si le paramètre de consignation registration-parameter de l'action automatique contient un paramètre submission-options, cette valeur est utilisée directement. Sinon les valeurs par défaut de dépôt générales (non propres à l'agent utilisateur UA) préalablement consignées à l'aide de l'opération abstraite Register-MS (consignation MS) sont utilisées;
- b) pour les actions automatiques qui entraînent la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques), une erreur dans l'opération abstraite MS-message-submission (dépôt de message MS) aura pour conséquence que la mémoire de messages attachera un attribut auto-action-error (erreur d'action automatique) indiquant la nature de l'erreur à l'entrée de la classe Auto-action-log. De plus, la mémoire de messages présentera une indication d'erreur d'action automatique à l'utilisateur MS dans le résultat de rattachement MS lors de l'établissement de l'association abstraite suivante.

15.2.2 Invocation de l'opération abstraite Probe-submission (dépôt de message d'essai)

L'invocation de l'opération abstraite Probe-submission (dépôt de message d'essai) résulte de l'invocation de l'opération abstraite MS-probe-submission (dépôt de message d'essai MS) par l'utilisateur MS. La définition de l'exécution de l'opération MS-probe-submission inclut une description de l'invocation de l'opération abstraite Probe-submission; voir 16.2.2.

15.2.3 Invocation de l'opération abstraite Cancel-deferred-delivery (annulation de remise différée)

Lorsque l'utilisateur MS invoque l'opération abstraite Cancel-deferred-delivery (annulation de remise différée), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages invoque l'opération abstraite Cancel-deferred-delivery (annulation de remise différée) à travers son association abstraite avec le système MTS, en utilisant l'argument d'annulation de remise différée fourni par l'utilisateur MS; voir 8.2.1.3.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) si le système MTS renvoie un résultat (indiquant l'aboutissement) et si l'action automatique Auto-correlate-reports (rapports de corrélation automatique) fait l'objet d'un abonnement, la mémoire de messages cherche dans les classes d'entrée Submission (dépôt) et Submission-log (journal de dépôt) une entrée

correspondant au message déposé pour lequel la remise différée a été annulée. Si une telle entrée est trouvée, la mémoire de messages lui attache alors un attribut `deferred-delivery-cancellation-time` (heure d'annulation de la remise différée) pour enregistrer la date et l'heure auxquelles l'annulation a eu lieu. L'attribut `AC-report-summary` (récapitulation des rapports AC) est également mis à jour;

- c) si l'exécution de l'opération abstraite aboutit, le résultat est renvoyé à l'utilisateur MS sous forme d'un résultat d'annulation de remise différée émis par la mémoire de messages. Sinon l'erreur est renvoyée à l'utilisateur MS.

15.2.4 Fonctionnement de l'opération abstraite `Submission-control` (commande de dépôt)

Quand le système MTS invoque l'opération abstraite `Submission-control` (commande de dépôt), la mémoire de messages effectue les opérations suivantes:

- a) s'il n'existe pas d'association abstraite entre la mémoire de messages et l'utilisateur MS, une erreur de rattachement distant est renvoyée au système MTS et la procédure prend fin;
- b) la mémoire de messages invoque une opération abstraite `Submission-control` (commande de dépôt) à travers son association abstraite avec l'utilisateur MS en utilisant l'argument de commande de dépôt fourni par le système MTS;
- c) l'utilisateur MS peut renvoyer un résultat pour indiquer que les commandes spécifiées sont maintenant en vigueur; sinon une erreur est renvoyée. Le résultat indique les opérations abstraites que l'utilisateur MS invoquerait, ou les types de message qu'il déposerait en l'absence des commandes prédominantes; voir 8.2.1.4.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- d) le résultat ou l'erreur renvoyé par l'utilisateur MS est renvoyé au système MTS;
- e) lorsque le système MTS est en mesure d'accepter le dépôt de messages ou de messages d'essai, il peut invoquer l'opération abstraite `Submission-control` (commande de dépôt) pour annuler les restrictions. Les effets d'une opération `Submission-control` sont annulés lorsqu'une nouvelle opération `Submission-control` modifie les restrictions ou lorsque l'association abstraite entre la mémoire de messages et le système MTS est libérée.

15.3 Utilisation des services abstraits de l'accès d'administration

Ce paragraphe décrit le fonctionnement des opérations abstraites `Register` (consignation) et `Change-credentials` (modification des pouvoirs). L'utilisation des services abstraits de l'accès d'administration suppose qu'une association abstraite existe entre le fournisseur de l'accès d'administration (le système MTS) et le client de l'accès d'administration (la mémoire de messages). Les opérations abstraites sont exécutées séquentiellement et aucun traitement parallèle n'a lieu. Les cas d'erreur ne sont pas décrits.

L'utilisation par la mémoire de messages de l'accès d'administration est soumise à la politique de sécurité en vigueur.

15.3.1 Invocation de l'opération abstraite `Register` (consignation)

Lorsque l'utilisateur MS invoque l'opération abstraite `Register` (consignation), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages invoque l'opération abstraite `Register` (consignation) à travers son association abstraite avec le système MTS, en utilisant l'argument de consignation fourni par l'utilisateur MS; voir 8.4.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) le résultat ou l'erreur renvoyé par le système MTS est renvoyé à l'utilisateur MS.

NOTE – Certaines politiques de sécurité peuvent permettre de modifier les étiquettes de sécurité utilisateur uniquement si une liaison sécurisée est utilisée.

15.3.2 Invocation de l'opération abstraite `Change-credentials` (modification des pouvoirs)

Lorsque l'utilisateur MS invoque l'opération abstraite `Change-credentials` (modification des pouvoirs), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages invoque l'opération abstraite `Change-credentials` (modification des pouvoirs) à travers son association abstraite avec le système MTS, en utilisant l'argument de modification des pouvoirs fourni par l'utilisateur MS; voir 8.4.1.2.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) le système MTS crée une erreur ou un résultat de modification des pouvoirs, que la mémoire de messages renvoie à l'utilisateur MS. Si l'opération abstraite a abouti, la mémoire de messages enregistre les nouveaux pouvoirs.

15.3.3 Fonctionnement de l'opération abstraite Change-credentials (modification des pouvoirs)

Lorsque le système MTS invoque l'opération abstraite Change-credentials (modification des pouvoirs), le système effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Change-credentials (modification des pouvoirs); voir 8.4.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Si les anciens pouvoirs sont incorrects ou si les nouveaux pouvoirs sont inacceptables, une erreur est renvoyée et la procédure prend fin;
- b) la mémoire de messages enregistre les nouveaux pouvoirs qui seront utilisés lors de ses rattachements ultérieurs au système MTS.

16 Fourniture du service abstrait de mémoire de messages

Cet article indique comment la mémoire de messages fournit le service abstrait MS. Il décrit la fourniture des services abstraits disponibles aux accès d'extraction, de dépôt MS et d'administration.

16.1 Fourniture des services abstraits de l'accès d'extraction

Ce paragraphe décrit la fourniture des opérations abstraites Summarize (récapitulation), List (listage), Fetch (recherche), Delete (suppression), Register MS (consignation MS), Modify (modification) et Alert (avertissement). La fourniture par le service abstrait MS des services abstraits de l'accès d'extraction suppose qu'une association abstraite existe entre le fournisseur de l'accès d'extraction (la mémoire de messages) et le client de l'accès d'extraction (l'utilisateur MS). L'exécution des opérations abstraites est asynchrone et soumise aux conditions déclarées au 8.2. Tous les cas d'erreur ne sont pas décrits.

16.1.1 Fonctionnement de l'opération abstraite Summarize (récapitulation)

Lorsque l'utilisateur MS invoque l'opération abstraite Summarize (récapitulation), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Summarize (récapitulation). Tous les types d'attribut spécifiés dans le paramètre summary-requests (demandes de récapitulation) doivent être disponibles pour être utilisés avec l'opération Summarize et l'utilisateur MS doit y être abonné; voir 8.2.1.1 et le Tableau 2. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) la mémoire de messages établit quelle classe d'entrées est appelée par l'opération abstraite et identifie l'entrée ou les entrées spécifiées dans l'argument. Si aucune entrée n'est sélectionnée, cela est signalé à l'utilisateur MS dans le résultat de récapitulation et la procédure prend fin;
- c) si des entrées sont sélectionnées, la mémoire de messages totalise les décomptes conformément au paramètre summary-requests (demandes de récapitulation) fourni;
- d) la mémoire de messages renvoie le résultat de récapitulation à l'utilisateur MS; voir 8.2.1.2;
- e) si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique de sécurité n'est pas transgressée pendant l'opération abstraite Summarize. Si cette opération demandée est interdite par la politique de sécurité, elle doit être abandonnée et une erreur de sécurité doit être signalée.

16.1.2 Fonctionnement de l'opération abstraite List (listage)

Lorsque l'utilisateur MS invoque l'opération abstraite List (listage), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite List (listage). Tous les types d'attribut spécifiés dans le paramètre requested-attributes (attributs demandés) doivent être disponibles pour être utilisés avec l'opération List et l'utilisateur MS doit y être abonné; voir 8.2.2.1 et le Tableau 2. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) la mémoire de messages établit quelle classe d'entrées est appelée par l'opération abstraite et identifie l'entrée ou les entrées spécifiées dans l'argument. Si aucune entrée n'est sélectionnée, la mémoire de messages renvoie un résultat de listage à l'utilisateur MS et la procédure prend fin;
- c) si un paramètre requested-attributes (attributs demandés) est spécifié dans l'argument de listage, ces attributs (le cas échéant) sont renvoyés pour chaque entrée sélectionnée. Si aucun attribut demandé n'est spécifié dans l'argument de listage, la mémoire de messages détermine si la présente association abstraite

a identifié, dans son argument de rattachement MS, une consignation d'agent utilisateur qui spécifie des valeurs par défaut pour les attributs de listage de l'agent utilisateur. Si c'est le cas, ces attributs (s'ils sont présents) sont renvoyés pour chaque entrée sélectionnée. Sinon, la mémoire de messages renvoie les attributs (le cas échéant) spécifiés dans les valeurs par défaut générales (non propres à l'agent utilisateur) des attributs de listage si l'utilisateur MS les a préalablement consignées à l'aide de l'opération abstraite Register-MS (consignation MS); voir 8.2.5.1 c). La valeur du paramètre retrieval-status (état d'extraction) de chaque entrée Stored-message (message enregistré) pour laquelle le paramètre entry-information (information d'entrée) est renvoyé, est mise à *listé* si sa valeur courante est à *nouveau*. Des règles supplémentaires affectant le positionnement du paramètre retrieval-status peuvent être définies dans les Spécifications pour les types de contenu particuliers;

- d) la mémoire de messages renvoie un résultat de listage à l'utilisateur MS; voir 8.2.2.2;
- e) si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique de sécurité n'est pas transgressée pendant l'opération List (listage). Si cette opération demandée est interdite, soit par la politique de sécurité soit par des restrictions de sécurité temporaires, elle doit être abandonnée et une erreur de sécurité doit être signalée.

16.1.3 Fonctionnement de l'opération abstraite Fetch (recherche)

Lorsque l'utilisateur MS invoque l'opération abstraite Fetch (recherche), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Fetch (recherche). L'utilisateur MS doit être abonné à tous les types d'attribut spécifiés dans le paramètre requested-attributes (attributs demandés); voir 8.2.3.1 et le Tableau 2. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) la mémoire de messages établit quelle classe d'entrées est appelée par l'opération abstraite et identifie l'entrée ou les entrées spécifiées dans l'argument. Si aucune entrée n'est sélectionnée, la mémoire de messages renvoie un résultat de recherche à l'utilisateur MS et la procédure prend fin;
- c) les restrictions de recherche sur les types de contenu autorisés établies par l'opération abstraite MS-bind (rattachement MS) seront appliquées (si elles n'ont pas été ignorées) afin de déterminer si les attributs demandés de l'entrée sélectionnée doivent être renvoyés ou si une erreur en résulte; voir 7.1.1 d);
- d) si un paramètre requested-attributes (attributs demandés) est spécifié dans l'argument de recherche, ces attributs, s'ils sont présents, sont renvoyés pour la première entrée sélectionnée. Si aucun attribut demandé n'est spécifié dans l'argument de recherche, la mémoire de messages détermine si la présente association abstraite a identifié, dans son argument de rattachement MS, une consignation d'agent utilisateur qui spécifie des valeurs par défaut pour les attributs de recherche de l'agent utilisateur. Si c'est le cas, ces attributs (le cas échéant) sont renvoyés. Sinon, la mémoire de messages renvoie les attributs (le cas échéant) spécifiés dans les valeurs par défaut générales (non propres à l'agent utilisateur) des attributs de recherche si l'utilisateur MS les a préalablement consignées à l'aide de l'opération abstraite Register-MS (consignation MS); voir 8.2.5.1 d). Les restrictions de recherche sur les types d'information codée autorisés et sur la longueur maximale des attributs établies par l'argument de rattachement MS peuvent limiter les informations renvoyées; voir 7.1.1 d).

Si la première entrée sélectionnée appartient à la classe d'entrées Stored-message (message enregistré) et que son état d'extraction est à *nouveau*, cet état est positionné à *listé*. Des règles supplémentaires affectant le positionnement de l'état d'extraction peuvent être définies dans la Spécification concernant un type de contenu donné. Si plusieurs entrées répondent aux critères de recherche, les numéros de séquence de la deuxième entrée et des entrées suivantes sont renvoyés dans l'ordre spécifié par le sélecteur (voir 8.1.3). Si le nombre des entrées répondant aux critères est supérieur à la limite spécifiée, le numéro de séquence qui suit immédiatement la limite est également renvoyé; voir 8.2.3.2;

- e) la mémoire de messages renvoie le résultat de recherche à l'utilisateur MS;
- f) si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique de sécurité n'est pas transgressée pendant l'opération abstraite Fetch. Si cette opération demandée est interdite soit par la politique de sécurité, soit par des restrictions de sécurité temporaires, elle doit être abandonnée et une erreur de sécurité doit être signalée.

16.1.4 Fonctionnement de l'opération abstraite Delete (suppression)

Lorsque l'utilisateur MS invoque l'opération abstraite Delete (suppression), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Delete (suppression); voir 8.2.4.1. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) la mémoire de messages établit quelle classe d'entrées est appelée par l'opération abstraite et identifie l'entrée ou les entrées spécifiées dans l'argument. Si aucune entrée n'est sélectionnée, la mémoire de messages renvoie un résultat de suppression à l'utilisateur MS et la procédure prend fin;
- c) si on rencontre une entrée dont la suppression est interdite par une restriction de suppression (voir 8.2.4), une erreur est créée qui inclut une indication des numéros de séquence des entrées dont la suppression a abouti (le cas échéant) et la procédure prend fin. Au niveau local, la mémoire de messages peut vérifier qu'aucune restriction de suppression ne s'applique aux entrées sélectionnées avant de tenter de les supprimer. Si une entrée de la classe Stored-message (message enregistré) est supprimée, l'attribut deletion-time (heure de suppression) de l'entrée correspondante de la classe d'entrées Message-log (journal de messagerie) est positionné à l'heure à laquelle la suppression a eu lieu;
- d) la mémoire de messages renvoie un résultat de suppression à l'utilisateur MS; voir 8.2.4.2. Si l'argument de suppression contient un paramètre selector (sélecteur) pour spécifier les entrées qu'il faut supprimer, les numéros de séquence de ces entrées sont signalés dans le résultat de suppression.

16.1.5 Fonctionnement de l'opération abstraite Register-MS (consignation MS)

Lorsque l'utilisateur MS invoque l'opération abstraite Register-MS (consignation MS), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Register-MS (consignation MS); voir 8.2.5.1. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) les arguments fournis peuvent entraîner l'ajout, la suppression ou la modification de consignations. En général, la consignation d'une action automatique n'a pas d'effet sur les entrées existantes; seules les entrées créées après la consignation de l'action automatique sont soumises à l'exécution de cette action. Cependant, les actions automatiques dont l'exécution n'est pas déclenchée par la création d'une entrée [comme l'action Auto-delete (suppression automatique), qui est déclenchée par l'expiration d'une temporisation] auront un effet sur les entrées existantes;
- c) si les arguments fournis incluent une demande d'état de consignation, les informations demandées sont renvoyées à l'utilisateur MS. Sinon, un résultat néant est renvoyé;
- d) si une politique de sécurité est en vigueur, l'opération abstraite Register-MS devra la respecter. Selon certaines politiques de sécurité, on ne peut modifier les étiquettes de sécurité de l'utilisateur que si on utilise une liaison sécurisée. Il existe d'autres moyens locaux pour modifier de façon sécurisée les étiquettes de sécurité de l'utilisateur.

16.1.6 Fonctionnement de l'opération abstraite Modify (modification)

Lorsque l'utilisateur MS invoque l'opération abstraite Modify (modification), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Modify (modification). La mémoire de messages essaie de détecter toutes les erreurs statiques possibles avant d'appliquer les modifications demandées; voir 8.2.7.1. Si une erreur est trouvée, la procédure prend fin et l'erreur est renvoyée;
- b) la mémoire de messages établit quelle classe d'entrées est appelée par l'opération abstraite et identifie l'entrée ou les entrées spécifiées dans l'argument;
- c) la mémoire de messages applique la séquence de modifications dans l'ordre spécifié à chaque entrée sélectionnée tour à tour. Si une erreur est rencontrée, la procédure prend fin en abandonnant l'entrée qui a entraîné l'erreur. L'erreur de modification indique les numéros de séquence des entrées dont la modification a abouti ainsi que le numéro de séquence de l'entrée qui a entraîné l'erreur;
- d) le résultat est renvoyé à l'utilisateur MS; il contient une liste des numéros de séquence des entrées sélectionnées à modifier (le cas échéant).

16.1.7 Invocation de l'opération abstraite Alert (avertissement)

Lorsque le système MTS invoque les opérations abstraites Message-delivery (remise de message) ou Report-delivery (remise de rapport), la mémoire de messages effectue les opérations suivantes:

- a) chaque action automatique Auto-alert (avertissement automatique) consignée est exécutée jusqu'à ce que l'une d'elles réussisse à donner l'avertissement ou jusqu'à ce que toutes aient été essayées. Si le message ou le rapport remis est conforme aux critères de sélection consignés et que le paramètre requested-attributes (attributs demandés) est présent [ou que la composante alert-destinations (destinations d'avertissement) est absente], la mémoire de messages invoque l'opération abstraite Alert. (A noter que cette procédure, de même que les autres définies au 16.1, suppose qu'une association abstraite existe entre la mémoire de messages et l'utilisateur MS; s'il n'existe pas d'association abstraite, l'opération Alert ne peut pas être invoquée.) Si l'exécution de l'opération Alert aboutit, l'utilisateur MS renvoie un résultat d'avertissement;
- b) si la composante alert-destinations (destinations d'avertissement) est présente [et si l'exécution de l'opération Alert a échoué, ou si la composante suppress-alert-destinations (suppression des destinations d'avertissement) est à *faux*], la mémoire de messages tente de donner l'avertissement aux destinations d'avertissement chacune à leur tour, jusqu'à ce qu'une tentative aboutisse, comme cela est défini au 13.1;
- c) si, après le traitement de toutes les consignations de l'opération Auto-alert, l'exécution de l'opération abstraite Alert n'a pas abouti, et si aucune destination d'avertissement n'a été avertie, la mémoire de messages positionne l'indication d'avertissement qui est signalée à l'utilisateur MS lors de l'établissement de l'association abstraite suivante;
- d) si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique ne sera pas transgressée pendant l'opération Alert. Si cette opération est interdite, soit par la politique de sécurité, soit par des restrictions de sécurité temporaires, la mesure à prendre doit respecter la politique de sécurité en vigueur.

16.2 Fourniture des services abstraits de l'accès de dépôt MS

Ce paragraphe décrit le fonctionnement des opérations abstraites MS-message-submission (dépôt de message MS), MS-probe-submission (dépôt de message d'essai MS) et MS-cancel-deferred-delivery (annulation de remise différée MS) ainsi que l'invocation de l'opération abstraite Submission-control (commande de dépôt). La fourniture, par le service abstrait MS, des services abstraits de l'accès de dépôt MS suppose qu'il existe une association abstraite entre le fournisseur de l'accès de dépôt MS (la mémoire de messages) et le client de l'accès de dépôt MS (l'utilisateur MS). Ces opérations abstraites sont exécutées séquentiellement et aucun traitement parallèle n'a lieu. Tous les cas d'erreur ne sont pas décrits.

16.2.1 Fonctionnement de l'opération abstraite MS-message-submission (dépôt de message MS)

Lorsque l'utilisateur MS invoque l'opération abstraite MS-message-submission (dépôt de message MS), la mémoire de messages effectue les opérations suivantes:

- a) si un contexte d'application 1994 est utilisé, le paramètre submission-options (options de dépôt) de l'argument de dépôt de message MS est extrait et l'argument de dépôt de message MTS est construit à partir de ce qui reste. Si le paramètre submission-options est absent et si la présente association abstraite a identifié une consignation d'agent utilisateur dans son argument de rattachement MS, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt de l'agent utilisateur concernant cette consignation d'agent utilisateur. Si la présente association abstraite n'a pas identifié de consignation d'agent utilisateur, ou que la consignation d'agent utilisateur ne contenait pas de valeur par défaut de dépôt d'agent utilisateur, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) consignées par l'opération Register-MS (consignation MS). Si les options de dépôt contiennent des extensions de dépôt MS, celles-ci sont sollicitées conformément aux règles établies dans la Spécification qui définit le type de contenu du message déposé. Si la mémoire de messages n'est pas en mesure d'effectuer le dépôt parce que les options de dépôt sont mal spécifiées ou qu'elles incluent des extensions de dépôt MS qui ne sont pas prises en charge, la mémoire de messages créera une entrée dans la classe d'entrées Submission-log (journal de dépôt) et lui attachera une erreur de dépôt MS; dans ce cas, la procédure continue à l'étape g). Sinon, elle continue à l'étape c).

Si un contexte d'application 1988 est utilisé, mais que l'utilisateur MS a consigné des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) à l'aide de l'opération Register-MS (consignation MS) lorsqu'il a utilisé un contexte d'application 1994, le paramètre submission-options (options de dépôt) de l'argument de dépôt de message MS est obtenu à partir de cette consignation;

- b) si un contexte d'application 1988 est utilisé, et que l'argument de dépôt de message MS contient un paramètre forwarding-request (demande de retransmission), la mémoire de messages vérifie que l'entrée à retransmettre est une entrée delivered-message (message remis) de la classe d'entrées Delivery (remise) et intègre l'entrée dans l'argument de dépôt de message MTS conformément aux règles propres au type de contenu;
- c) si les options de dépôt contenaient une demande de création d'une entrée draft-message (brouillon de message) (sans dépôt auprès du système MTS), cette entrée est créée et la procédure continue à l'étape g) ci-dessous. Les types d'attribut obligatoires et optionnels pour les entrées de la classe Draft (brouillon) sont indiqués dans le Tableau 2;
- d) la mémoire de messages essaie d'établir une association abstraite avec le système MTS s'il n'en existe pas déjà une. S'il est impossible d'établir une association abstraite, une erreur de rattachement distant est renvoyée à l'utilisateur MS et la procédure prend fin;
- e) la mémoire de messages crée une entrée dans la classe d'entrées Submission-log (journal de dépôt) et invoque l'opération abstraite Message-submission (dépôt de message) à travers son association abstraite avec le système MTS. Les types d'attribut obligatoires et optionnels pour les entrées submitted-message (message déposé) de la classe d'entrées Submission-log (journal de dépôt) sont indiqués dans le Tableau 2. Si l'exécution de l'opération Message-submission échoue, la mémoire de messages attache un attribut MS-submission-error (erreur de dépôt MS) à l'entrée de la classe Submission-log pour enregistrer l'erreur et la procédure continue à l'étape g);
- f) si l'exécution de l'opération Message-submission aboutit et que le paramètre submission-options (options de dépôt) (ou sa valeur par défaut consignée) a demandé la création d'une entrée dans la classe d'entrées Submission (dépôt), cette entrée est créée. Les types d'attribut obligatoires et optionnels pour les entrées submitted-message (message déposé) de la classe d'entrées Submission (dépôt) sont indiqués dans le Tableau 2. Si l'utilisateur MS est abonné à l'action automatique Auto-correlate-reports (rapports de corrélation automatique), la mémoire de messages crée les attributs de corrélation indiqués au 13.3;
- g) l'entrée créée dans la classe d'entrées Submission-log (journal de dépôt) ou Draft (brouillon), ou les entrées créées à la fois dans la classe Submission-log et dans la classe Submission sont modifiées de la façon suivante. Si le paramètre submission-options (options de dépôt) (ou sa valeur par défaut consignée) a spécifié un ou plusieurs noms de groupe de messages à ajouter à la ou aux nouvelles entrées, ces valeurs d'attribut sont alors ajoutées. Si des actions automatiques Auto-modify (modification automatique) ont été consignées par l'utilisateur MS et autorisées dans le paramètre submission-options (ou dans sa valeur par défaut), chacune est examinée à son tour. Si l'entrée appartient à la classe d'entrées spécifiée pour cette action automatique consignée et qu'elle est conforme aux critères de sélection spécifiés, la séquence de modifications est appliquée. Si des actions automatiques propres au contenu ont été consignées, elles sont exécutées.

Certaines actions automatiques, lorsqu'elles sont exécutées, entraînent la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques). Si l'une quelconque de ces actions automatiques entraîne une erreur d'action automatique, la mémoire de messages attachera un attribut auto-action-error (erreur d'action automatique) indiquant la nature de l'erreur à l'entrée de la classe Auto-action-log et positionnera l'indication d'erreur d'action automatique, qui est renvoyée à l'utilisateur MS dans le résultat de dépôt de message MS. La mémoire de messages continue alors l'exécution des actions automatiques consignées.

Si l'exécution de l'opération Message-submission échoue [ou a échoué à l'étape a) ci-dessus], l'erreur est signalée à l'utilisateur MS et la procédure continue à l'étape i);

- h) le résultat de dépôt de message MS est renvoyé à l'utilisateur MS. Il doit indiquer le numéro de séquence de toute entrée principale créée dans la classe d'entrées Draft (brouillon), Submission (dépôt) ou Submission-log (journal de dépôt). Si l'opération Message-submission (dépôt de message) a été demandée (et non l'enregistrement d'un brouillon de message), le résultat de dépôt de message MS inclura le résultat de dépôt de message MTS (voir 8.2.1.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4);
- i) la mémoire de messages peut choisir de mettre fin à l'association abstraite avec le système MTS, soit lorsque l'utilisateur MS met fin à son association abstraite avec la mémoire de messages, soit après une certaine période d'inactivité;
- j) si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique ne sera pas transgressée pendant l'opération MS-message-submission (dépôt de message MS). Si cette opération est interdite soit par la politique de sécurité, soit par des restrictions de sécurité temporaires, une erreur de sécurité doit être signalée.

16.2.2 Fonctionnement de l'opération abstraite MS-probe-submission (dépôt de message d'essai MS)

Lorsque l'utilisateur MS invoque l'opération abstraite MS-probe-submission (dépôt de message d'essai MS), la mémoire de messages effectue les opérations suivantes:

- a) si un contexte d'application 1994 est utilisé, le paramètre submission-options (options de dépôt) de l'argument de dépôt de message d'essai MS est extrait et l'argument de dépôt de message d'essai MTS est construit à partir de ce qui reste. Si le paramètre submission-options est absent et si la présente association abstraite a identifié une consignation d'agent utilisateur dans son argument de rattachement MS, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt de l'agent utilisateur concernant cette consignation d'agent utilisateur. Si la présente association abstraite n'a pas identifié de consignation d'agent utilisateur, ou que la consignation d'agent utilisateur ne contenait pas de valeur par défaut de dépôt d'agent utilisateur, les options de dépôt sont obtenues à partir des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) consignées par l'opération Register-MS (consignation MS). Si la mémoire de messages n'est pas en mesure d'effectuer le dépôt parce que les options de dépôt sont mal spécifiées, elle créera une entrée dans la classe d'entrées Submission-log (journal de dépôt) et lui attachera une erreur de dépôt MS; dans ce cas, la procédure continue à l'étape e).

Si un contexte d'application 1988 est utilisé, mais que l'utilisateur MS a consigné des valeurs par défaut de dépôt générales (non propres à l'agent utilisateur) à l'aide de l'opération Register-MS (consignation MS) lorsqu'il a utilisé un contexte d'application 1994, le paramètre submission-options (options de dépôt) de l'argument de dépôt de message d'essai MS est obtenu à partir de cette consignation;

- b) la mémoire de messages essaie d'établir une association abstraite avec le système MTS, s'il n'en existe pas déjà une. S'il est impossible d'établir une association abstraite, une erreur de rattachement distant est renvoyée à l'utilisateur MS et la procédure prend fin;
- c) la mémoire de messages crée une entrée dans la classe d'entrées Submission-log (journal de dépôt) et invoque l'opération abstraite Probe-submission (dépôt de message d'essai) à travers son association abstraite avec le système MTS. Les types d'attribut obligatoires et optionnels pour les entrées submitted-probe (message d'essai déposé) de la classe d'entrées Submission-log (journal de dépôt) sont indiqués dans le Tableau 2. Si l'exécution de l'opération Probe-submission échoue, la mémoire de messages attache un attribut MS-submission-error (erreur de dépôt MS) à l'entrée de la classe Submission-log pour enregistrer l'erreur et la procédure continue à l'étape e);
- d) si l'exécution de l'opération Probe-submission aboutit et que le paramètre submission-options (options de dépôt) (ou sa valeur par défaut consignée) a demandé la création d'une entrée dans la classe d'entrées Submission (dépôt), cette entrée est créée. Les types d'attribut obligatoires et optionnels pour les entrées submitted-probe (message d'essai déposé) de la classe d'entrées Submission (dépôt) sont indiqués dans le Tableau 2. Si l'utilisateur MS est abonné à l'action automatique Auto-correlate-reports (rapports de corrélation automatique), la mémoire de messages crée les attributs de corrélation indiqués au 13.3;
- e) l'entrée créée dans la classe d'entrées Submission-log (journal de dépôt), ou les entrées créées à la fois dans la classe Submission-log et dans la classe Submission sont modifiées de la façon suivante. Si le paramètre submission-options (options de dépôt) (ou sa valeur par défaut consignée) a spécifié un ou plusieurs noms de groupe de messages à ajouter à la ou aux nouvelles entrées, ces valeurs d'attribut sont alors ajoutées. Si des actions automatiques Auto-modify (modification automatique) ont été consignées par l'utilisateur MS et autorisées dans le paramètre submission-options (ou dans sa valeur par défaut), chacune est examinée à son tour. Si l'entrée appartient à la classe d'entrées spécifiée pour cette action automatique consignée et qu'elle est conforme aux critères de sélection spécifiés, la séquence de modifications est appliquée.

Certaines actions automatiques, lorsqu'elles sont exécutées, entraînent la création d'une entrée dans la classe d'entrées Auto-action-log (journal d'actions automatiques). Si l'une quelconque de ces actions automatiques entraîne une erreur d'action automatique, la mémoire de messages attachera un attribut auto-action-error (erreur d'action automatique) indiquant la nature de l'erreur à l'entrée de la classe Auto-action-log et positionnera l'indication d'erreur d'action automatique, qui est renvoyée à l'utilisateur MS dans le résultat de dépôt de message d'essai MS. La mémoire de messages continue alors l'exécution des actions automatiques consignées.

Si l'exécution de l'opération Probe-submission échoue [ou a échoué à l'étape a) ci-dessus], l'erreur est signalée à l'utilisateur MS et la procédure continue à l'étape g);

- f) le résultat de dépôt de message d'essai MS est renvoyé à l'utilisateur MS. Il doit indiquer le numéro de séquence de toute entrée créée dans les classes d'entrée Submission (dépôt) et Submission-log (journal de dépôt). Le résultat de dépôt de message d'essai MS inclura le résultat de dépôt de message d'essai MTS (voir 8.2.1.2.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4);

ISO/CEI 10021-5 : 1995 (F)

- g) la mémoire de messages peut choisir de mettre fin à l'association abstraite avec le système MTS, soit lorsque l'utilisateur MS met fin à son association abstraite avec la mémoire de messages, soit après une certaine période d'inactivité;
- h) Si une politique de sécurité est en vigueur, la mémoire de messages compare l'étiquette de sécurité du message au contexte de sécurité pour s'assurer que cette politique ne sera pas transgressée pendant l'opération MS-probe-submission (dépôt de message d'essai MS). Si cette opération est interdite soit par la politique de sécurité, soit par des restrictions de sécurité temporaires, une erreur de sécurité doit être signalée.

16.2.3 Fonctionnement de l'opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS)

Lorsque l'utilisateur MS invoque l'opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour une opération abstraite MS-cancel-deferred-delivery (annulation de remise différée MS); voir 8.2.1.3.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- b) la mémoire de messages tente d'établir une association abstraite avec le système MTS, s'il n'en existe pas déjà une. S'il est impossible d'établir une association abstraite, une erreur de rattachement distant est renvoyée à l'utilisateur MS et la procédure prend fin;
- c) la mémoire de messages fait appel au système MTS pour une opération abstraite Cancel-deferred-delivery. Si la demande d'annulation de remise aboutit et que l'utilisateur est abonné à l'élément de service de corrélation automatique des rapports, la mémoire de messages cherche dans les classes d'entrée Submission (dépôt) et Submission-log (journal de dépôt) une entrée correspondant au message déposé pour lequel une remise différée a été annulée. Si cette entrée est présente, la mémoire de messages lui attache un attribut deferred-delivery-cancellation-time (heure d'annulation de remise différée) pour enregistrer la date et l'heure auxquelles l'annulation de remise a eu lieu et met à jour l'attribut AC-report-summary (récapitulation des rapports AC) pour enregistrer l'annulation;
- d) la mémoire de messages signale le résultat de l'opération abstraite à l'utilisateur MS;
- e) la mémoire de messages peut choisir de mettre fin à l'association abstraite avec le système MTS, soit lorsque l'utilisateur MS met fin à son association abstraite avec la mémoire de messages, soit après une certaine période d'inactivité.

16.2.4 Invocation de l'opération abstraite Submission-control (commande de dépôt)

Si le système MTS invoque l'opération abstraite Submission-control (commande de dépôt) ou si, pour divers motifs internes, la mémoire de messages nécessite une restriction temporaire sur les opérations abstraites de dépôt que l'utilisateur MS peut invoquer ou si elle nécessite une modification des restrictions existantes, la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages invoque l'opération abstraite MS-submission-control (commande de dépôt MS); voir 8.3.4;
- b) la mémoire de messages attend que l'utilisateur MS crée un résultat de commande de dépôt MS confirmant l'acceptation de l'opération abstraite MS-submission-control;
- c) si l'opération Submission-control a été invoquée au départ par le système MTS, la mémoire de messages renvoie le résultat de commande de dépôt MS qu'elle a reçu de l'utilisateur MS au système MTS.

16.2.5 Règles de création des attributs généraux

Les entrées des classes d'entrée Submission (dépôt), Submission-log (journal de dépôt) et Draft (brouillon) créées comme résultat des opérations MS-message-submission (dépôt de message MS) et MS-probe-submission (dépôt de message d'essai MS) doivent contenir les attributs obligatoires indiqués dans le Tableau 2, et les attributs optionnels pris en charge par la réalisation de la mémoire de messages et auxquels l'utilisateur MS s'est abonné. Les attributs créés forment une nouvelle entrée, ou dans certains cas, une entrée mère et une ou plusieurs entrées filles; voir 6.3.4. Chaque attribut d'une entrée dans la classe d'entrées Submission-log a la même valeur que l'attribut équivalent dans l'entrée

correspondante de la classe d'entrées Submission [à l'exception des attributs deletion-time (heure de suppression) et MS-submission-error (erreur de dépôt MS) qui sont propres aux entrées de la classe Submission-log]. Les types suivants d'attributs généraux doivent être créés:

- a) attributs généraux créés par la mémoire de messages elle-même (par exemple le numéro de séquence);
- b) attributs généraux créés à partir des composantes des attributs message-submission-enveloppe (enveloppe de dépôt de message) et probe-submission-enveloppe (enveloppe de dépôt de message d'essai) et à partir du résultat des opérations abstraites Message-submission (dépôt de message) et Probe-submission (dépôt de message d'essai) du système MTS. Pour les composantes qui ne sont pas présentes, mais pour lesquelles des valeurs par défaut sont définies, un attribut général contenant la valeur par défaut est créé.

Voir au Tableau 4 et au 11.4 les règles de création des attributs généraux. Les règles de création des attributs propres au contenu sont décrites dans la Spécification concernant le type de contenu en question [par exemple les attributs propres au système de messagerie de personne à personne (IPMS) sont décrits à l'article 19 de la Rec. UIT-T X.420 | ISO/CEI 10021-7].

16.3 Fourniture des services abstraits de l'accès d'administration

Ce paragraphe décrit le fonctionnement des opérations abstraites Register (consignation) et Change-credentials (modification des pouvoirs). La fourniture par le service abstrait MS des services abstraits de l'accès d'administration suppose qu'il existe une association abstraite entre le fournisseur de l'accès d'administration (la mémoire de messages) et le client de l'accès d'administration (l'utilisateur MS). Ces opérations abstraites sont exécutées séquentiellement et aucun traitement parallèle n'a lieu. Tous les cas d'erreur ne sont pas décrits.

16.3.1 Fonctionnement de l'opération abstraite Register (consignation)

Lorsque l'utilisateur MS invoque l'opération abstraite Register (consignation), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Register (consignation); voir 8.4.1.1.1 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Sous réserve de politique locale ou d'abonnement, la mémoire de messages peut imposer des restrictions supplémentaires sur les consignations qui peuvent être exécutées par l'utilisateur MS; si ces restrictions sont transgressées, une erreur abstraite est renvoyée à l'utilisateur MS et la procédure prend fin;
- b) la mémoire de messages essaie d'établir une association abstraite avec le système MTS, s'il n'en existe pas déjà une. S'il est impossible d'établir une association abstraite, une erreur de rattachement distant est renvoyée à l'utilisateur MS et la procédure prend fin;
- c) la mémoire de messages invoque l'opération abstraite Register (consignation) à travers son association abstraite avec le système MTS, contenant les arguments de l'opération abstraite d'origine;
- d) le résultat ou l'erreur renvoyé par le système MTS est renvoyé à l'utilisateur MS;
- e) la mémoire de messages peut choisir de mettre fin à l'association abstraite avec le système MTS, soit lorsque l'utilisateur MS met fin à son association abstraite avec la mémoire de messages, soit après une certaine période d'inactivité;
- f) l'importance des modifications que l'utilisateur MS peut apporter aux étiquettes de sécurité de l'utilisateur devra respecter la politique de sécurité en vigueur. Selon certaines politiques de sécurité, on ne peut modifier les étiquettes de sécurité de l'utilisateur que si on utilise une liaison sécurisée. Il existe d'autres moyens locaux pour modifier de façon sécurisée les étiquettes de sécurité de l'utilisateur.

16.3.2 Invocation de l'opération abstraite Change-credentials (modification des pouvoirs)

Lorsque le système MTS invoque l'opération abstraite Change-credentials (modification des pouvoirs), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Change-credentials (modification des pouvoirs); voir 8.4.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Si les anciens pouvoirs sont incorrects ou si les nouveaux sont inacceptables, une erreur est renvoyée et la procédure prend fin;
- b) la mémoire de messages enregistre les nouveaux pouvoirs qui seront utilisés lors de ses prochains rattachements au système MTS et renvoie un résultat au système MTS.

16.3.3 Fonctionnement de l'opération abstraite Change-credentials (modification des pouvoirs)

Lorsque l'utilisateur MS invoque l'opération abstraite Change-credentials (modification des pouvoirs), la mémoire de messages effectue les opérations suivantes:

- a) la mémoire de messages vérifie que les arguments fournis sont valables pour l'opération abstraite Change-credentials (modification des pouvoirs); voir 8.4.1.2 de la Rec. UIT-T X.411 | ISO/CEI 10021-4. Sous réserve de politique locale, la mémoire de messages peut imposer des restrictions supplémentaires sur l'utilisation de l'opération abstraite Change-credentials par l'utilisateur MS; si ces restrictions sont transgressées, une erreur abstraite est renvoyée à l'utilisateur MS et la procédure prend fin;
- b) la mémoire de messages essaie d'établir une association abstraite avec le système MTS, s'il n'en existe pas déjà une. S'il est impossible d'établir une association abstraite, une erreur de rattachement distant est renvoyée à l'utilisateur MS et la procédure prend fin;
- c) la mémoire de messages invoque l'opération abstraite Change-credentials à travers son association abstraite avec le système MTS;
- d) le résultat ou l'erreur renvoyé par le système MTS est renvoyé à l'utilisateur MS. Si l'exécution de l'opération abstraite a abouti, la mémoire de messages enregistre les nouveaux pouvoirs;
- e) la mémoire de messages peut choisir de mettre fin à l'association abstraite avec le système MTS, soit lorsque l'utilisateur MS met fin à son association abstraite avec la mémoire de messages, soit après une certaine période d'inactivité.

17 Réalisation des accès

Cet article décrit comment les accès d'extraction, de dépôt MS et d'administration du service abstrait MS sont fournis. On trouvera à l'article 8 de la Rec. UIT-T X.411 | ISO/CEI 10021-4 une description de la fourniture par le service abstrait MTS des accès de remise, de dépôt et d'administration.

17.1 Accès d'extraction

Les services abstraits de l'accès d'extraction sont fournis un à un entre les opérations abstraites et les opérations réelles de l'élément de service d'extraction de messages, version 1988 (MRSE-88) et de l'élément de service d'extraction de messages, version 1994 (MRSE-94) qui sont définis dans la Rec. UIT-T X.419 | ISO/CEI 10021-6.

NOTE – L'élément MRSE-88 s'applique lors de l'utilisation d'un contexte d'application 1988; l'élément MRSE-94 s'applique lors de l'utilisation d'un contexte d'application 1994.

17.2 Accès de dépôt MS

Les services abstraits de l'accès de dépôt MS sont fournis un à un entre les opérations abstraites et les opérations réelles de l'élément de service de dépôt de message (MSSE) et de l'élément de service de dépôt de message MS (MS-MSSE) qui sont définis dans la Rec. UIT-T X.419 | ISO/CEI 10021-6.

NOTE – L'élément MSSE s'applique lors de l'utilisation d'un contexte d'application 1988; l'élément MS-MSSE s'applique lors de l'utilisation d'un contexte d'application 1994.

17.3 Accès d'administration

Les services abstraits de l'accès d'administration sont fournis un à un entre les opérations abstraites et les opérations réelles de l'élément de service de gestion de messages, version 1988 (MASE-88) et de l'élément de service de gestion de messages, version 1994 (MASE-94) qui sont définis dans la Rec. UIT-T X.419 | ISO/CEI 10021-6.

NOTE – L'élément MASE-88 s'applique lors de l'utilisation d'un contexte d'application 1988; l'élément MASE-94 s'applique lors de l'utilisation d'un contexte d'application 1994.

Annexe A

Affectation formelle des identificateurs d'objets

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

Tous les identificateurs d'objets affectés dans la présente Définition de service sont affectés de façon formelle, dans la présente annexe, au moyen de la notation ASN.1. Les valeurs spécifiées sont indiquées dans les modules ASN.1 des annexes suivantes.

La présente annexe est définitive pour toutes les valeurs, à l'exception de celles qui sont applicables aux identificateurs des modules ASN.1 et à l'ensemble de la présente Définition de service. Les affectations définitives pour les modules apparaissent dans les modules eux-mêmes. Dans le second cas, les affectations sont fixes. D'autres références aux valeurs affectées aux modules apparaissent dans les clauses IMPORT.

MSubjectIdentifiers {joint-iso-itu-t mhs(6) ms(4) modules(0) object-identifiers(0) version-1994(0)};

DEFINITIONS ::=

BEGIN

-- Prologue

-- Exporte tout

IMPORTS

ID, id-ms

FROM MSubjectIdentifiers {joint-iso-itu-t mhs(6) arch(5) modules(0) object-identifiers(0) version-1994(0)};

-- Catégories

id-mod	-- modules --	ID ::= {id-ms 0}
id-ot	-- objets --	ID ::= {id-ms 1}
id-pt	-- types d'accès --	ID ::= {id-ms 2}
id-att	-- types d'attribut --	ID ::= {id-ms 3}
id-act	-- types d'action automatique --	ID ::= {id-ms 4}
id-crt	-- contrats --	ID ::= {id-ms 5}
id-cp	-- blocs de connexions --	ID ::= {id-ms 6}
id-aae	-- erreurs d'action automatique --	ID ::= {id-ms 7}
id-mr	-- règles de criblage --	ID ::= {id-ms 8}

-- Modules

id-mod-object-identifiers	ID ::= {id-mod 0} -- affectation non définitive
id-mod-abstract-service	ID ::= {id-mod 1} -- affectation non définitive
id-mod-attribute-types	ID ::= {id-mod 2} -- affectation non définitive
id-mod-action-types	ID ::= {id-mod 3} -- affectation non définitive
id-mod-upper-bounds	ID ::= {id-mod 4} -- affectation non définitive
id-mod-matching-rules	ID ::= {id-mod 5} -- affectation non définitive

-- Objets

id-ot-ms	ID ::= {id-ot 0}
id-ot-ms-user	ID ::= {id-ot 1}

-- Types d'accès

id-pt-retrieval-88	ID ::= {id-pt 0}
id-pt-retrieval-94	ID ::= {id-pt 1}
id-pt-ms-submission	ID ::= {id-pt 2}

-- Contrats

id-crt-ms-access-88	ID ::= {id-crt 0}
id-crt-ms-access-94	ID ::= {id-crt 1}

-- Blocs de connexions

id-cp-ms-connection	ID ::= {id-cp 0}
----------------------------	-------------------------

-- *Types d'attribut*

id-att-ac-correlated-report-list	ID ::= {id-att 42}
id-att-ac-report-summary	ID ::= {id-att 43}
id-att-ac-uncorrelated-report-list	ID ::= {id-att 44}
id-att-auto-action-error	ID ::= {id-att 46}
id-att-auto-action-registration-identifier	ID ::= {id-att 47}
id-att-auto-action-subject-entry	ID ::= {id-att 48}
id-att-auto-action-type	ID ::= {id-att 49}
id-att-child-sequence-numbers	ID ::= {id-att 0}
id-att-content	ID ::= {id-att 1}
id-att-content-confidentiality-algorithm-identifier	ID ::= {id-att 2}
id-att-content-correlator	ID ::= {id-att 3}
id-att-content-identifier	ID ::= {id-att 4}
id-att-content-integrity-check	ID ::= {id-att 5}
id-att-content-length	ID ::= {id-att 6}
id-att-content-returned	ID ::= {id-att 7}
id-att-content-type	ID ::= {id-att 8}
id-att-conversion-with-loss-prohibited	ID ::= {id-att 9}
id-att-converted-EITs	ID ::= {id-att 10}
id-att-creation-time	ID ::= {id-att 11}
id-att-deferred-delivery-cancellation-time	ID ::= {id-att 50}
id-att-deferred-delivery-time	ID ::= {id-att 51}
id-att-deletion-time	ID ::= {id-att 52}
id-att-delivered-EITs	ID ::= {id-att 12}
id-att-delivery-flags	ID ::= {id-att 13}
id-att-dl-expansion-history	ID ::= {id-att 14}
id-att-dl-expansion-prohibited	ID ::= {id-att 53}
id-att-entry-type	ID ::= {id-att 16}
id-att-internal-trace-information	ID ::= {id-att 54}
id-att-latest-delivery-time	ID ::= {id-att 55}
id-att-marked-for-deletion	ID ::= {id-att 56}
id-att-message-delivery-envelope	ID ::= {id-att 18}
id-att-message-delivery-time	ID ::= {id-att 20}
id-att-message-group-name	ID ::= {id-att 57}
id-att-message-identifier	ID ::= {id-att 19}
id-att-message-notes	ID ::= {id-att 58}
id-att-message-origin-authentication-check	ID ::= {id-att 21}
id-att-message-security-label	ID ::= {id-att 22}
id-att-message-submission-envelope	ID ::= {id-att 59}
id-att-message-submission-time	ID ::= {id-att 23}
id-att-message-token	ID ::= {id-att 24}
id-att-ms-originated	ID ::= {id-att 60}
id-att-ms-submission-error	ID ::= {id-att 61}
id-att-original-EITs	ID ::= {id-att 25}
id-att-originally-intended-recipient-name	ID ::= {id-att 17}
id-att-originating-MTA-certificate	ID ::= {id-att 62}
id-att-originator-certificate	ID ::= {id-att 26}
id-att-originator-name	ID ::= {id-att 27}
id-att-originator-report-request	ID ::= {id-att 63}
id-att-originator-return-address	ID ::= {id-att 64}
id-att-other-recipient-names	ID ::= {id-att 28}
id-att-parent-sequence-number	ID ::= {id-att 29}
id-att-per-message-indicators	ID ::= {id-att 65}
id-att-per-recipient-message-submission-fields	ID ::= {id-att 66}
id-att-per-recipient-probe-submission-fields	ID ::= {id-att 67}
id-att-per-recipient-report-delivery-fields	ID ::= {id-att 30}
id-att-priority	ID ::= {id-att 31}
id-att-probe-origin-authentication-check	ID ::= {id-att 68}
id-att-probe-submission-envelope	ID ::= {id-att 69}
id-att-proof-of-delivery-request	ID ::= {id-att 32}
id-att-proof-of-submission	ID ::= {id-att 70}
id-att-recipient-names	ID ::= {id-att 71}
id-att-recipient-reassignment-prohibited	ID ::= {id-att 72}
id-att-redirection-history	ID ::= {id-att 33}
id-att-report-delivery-envelope	ID ::= {id-att 34}
id-att-reporting-DL-name	ID ::= {id-att 35}
id-att-reporting-MTA-certificate	ID ::= {id-att 36}
id-att-report-origin-authentication-check	ID ::= {id-att 37}
id-att-retrieval-status	ID ::= {id-att 15}

id-att-security-classification	ID ::= {id-att 38}
id-att-sequence-number	ID ::= {id-att 39}
id-att-storage-period	ID ::= {id-att 73}
id-att-storage-time	ID ::= {id-att 74}
id-att-subject-submission-identifier	ID ::= {id-att 40}
id-att-this-recipient-name	ID ::= {id-att 41}
id-att-trace-information	ID ::= {id-att 75}

-- *Types d'action automatique*

id-act-ipm-auto-forward <i>ISO/CEI 10021-7</i>	ID ::= {id-act 0} -- <i>Utilisation réservée dans la Rec. UIT-T X.420 </i>
id-act-auto-alert	ID ::= {id-act 1}
id-act-auto-correlate-reports	ID ::= {id-act 2}
id-act-auto-delete	ID ::= {id-act 3}
id-act-auto-modify	ID ::= {id-act 4}

-- *Erreurs d'action automatique*

id-aae-auto-alert-error	ID ::= {id-aae 0}
--------------------------------	--------------------------

-- *Règles de criblage*

id-mr-content-correlator-match	ID ::= {id-mr 1}
id-mr-content-identifier-match	ID ::= {id-mr 2}
id-mr-ms-single-substring-list-elements-match	ID ::= {id-mr 3}
id-mr-ms-single-substring-list-match	ID ::= {id-mr 4}
id-mr-ms-single-substring-match	ID ::= {id-mr 5}
id-mr-ms-substrings-match	ID ::= {id-mr 6}
id-mr-msstring-case-sensitive-match	ID ::= {id-mr 7}
id-mr-msstring-list-elements-match	ID ::= {id-mr 8}
id-mr-msstring-list-match	ID ::= {id-mr 9}
id-mr-msstring-match	ID ::= {id-mr 10}
id-mr-msstring-ordering-match	ID ::= {id-mr 11}
id-mr-mts-identifier-match	ID ::= {id-mr 12}
id-mr-oraddress-elements-match	ID ::= {id-mr 13}
id-mr-oraddress-match	ID ::= {id-mr 14}
id-mr-oraddress-substring-elements-match	ID ::= {id-mr 15}
id-mr-orname-elements-match	ID ::= {id-mr 16}
id-mr-orname-match	ID ::= {id-mr 17}
id-mr-orname-single-element-match	ID ::= {id-mr 18}
id-mr-orname-substring-elements-match	ID ::= {id-mr 19}
id-mr-redirect-or-dl-expansion-elements-match	ID ::= {id-mr 20}
id-mr-redirect-or-dl-expansion-match	ID ::= {id-mr 21}
id-mr-redirect-or-dl-expansion-substring-elements-match	ID ::= {id-mr 22}
id-mr-redirect-reason-match	ID ::= {id-mr 23}
id-mr-value-count-match	ID ::= {id-mr 24}

END -- *fin des identificateurs d'objets MS*

Annexe B

Définition formelle du service abstrait de mémoire de messages

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

La présente annexe, qui complète la section 2, définit de façon formelle le service abstrait de mémoire de messages. Ce service utilise la notation ASN.1 et les classes d'objets informationnels MHS-OBJECT (objet du système de messagerie), PORT (accès), ABSTRACT-OPERATION (opération abstraite) et ABSTRACT-ERROR (erreur abstraite) définies dans la Rec. UIT-T X.411 | ISO/CEI 10021-4 et les classes d'objets informationnels CONTRACT (contrat) et CONNECTION-PACKAGE (bloc de connexions) définies dans la Rec. UIT-T X.880 | ISO/CEI 13712-1.

NOTE – L'utilisation des classes d'objets informationnels MHS-OBJECT, PORT, ABSTRACT-OPERATION et ABSTRACT-ERROR qui sont dérivées des classes d'objets informationnels ROS-OBJECT-CLASS (classe d'objet ROS), OPERATION-PACKAGE (bloc d'opérations), OPERATION et ERROR (erreur) du service des opérations distantes (ROS) n'implique pas que les opérations abstraites et les erreurs abstraites soient dans chaque instance invoquées et signalées au-delà de la limite entre systèmes ouverts. Toutefois, cela sera fait fréquemment. La Rec. UIT-T X.419 | ISO/CEI 10021-6 traite des modalités de cette opération.

MSAbstractService {joint-iso-itu-t mhs(6) ms(4) modules(0) abstract-service(1) version-1994(0)}
DEFINITIONS ::=

BEGIN*-- Prologue**-- Exporte tout***IMPORTS***-- Classes d'objets informationnels MTS***ABSTRACT-ERROR, ABSTRACT-OPERATION, EXTENSION, MHS-OBJECT, PORT,***-- Objets et accès MTS***administration, delivery, mts-user, submission,***-- Opérations abstraites et erreurs abstraites MTS***cancel-deferred-delivery, element-of-service-not-subscribed, inconsistent-request, new-credentials-unacceptable, old-credentials-incorrectly-specified, originator-invalid, recipient-improperly-specified, remote-bind-error, security-error, submission-control, submission-control-violated, unsupported-critical-function,***-- Types de données du service abstrait MTS***Credentials, InitiatorCredentials, MessageSubmissionArgument, MessageSubmissionResult, ORAddressAndOrDirectoryName, ProbeSubmissionArgument, ProbeSubmissionResult, ResponderCredentials, SecurityContext, SecurityLabel****FROM MTSAbstractService** {joint-iso-itu-t mhs(6) mts(3) modules(0) mts-abstract-service(1) version-1994(0)}*-- Accès 1988 du service abstrait MTS***administration-88**

FROM MTSAbstractService88 {joint-iso-itu-t mhs(6) mts(3) modules(0) mts-abstract-service(1) version-1988(1988)}

*-- Bornes supérieures du service abstrait MTS***ub-content-types, ub-encoded-information-types, ub-labels-and-redirections**

FROM MTSUpperBounds {joint-iso-itu-t mhs(6) mts(3) modules(0) upper-bounds(3)}

*-- Tableau des attributs MS***AttributeTable**

FROM MSGGeneralAttributeTypes {joint-iso-itu-t mhs(6) ms(4) modules(0) general-attribute-types(2) version-1994(0)}

-- Tableau des règles de criblage MS

MatchingRuleTable

```
----
FROM MSMatchingRules {joint-iso-itu-t mhs(6) ms(4) modules(0) general-matching-rules(5)}
```

-- Tableau des actions automatiques et tableau des erreurs d'action automatique MS

AutoActionTable, AutoActionErrorTable

```
----
FROM MSGeneralAutoActionTypes {joint-iso-itu-t mhs(6) ms(4) modules(0)
                               general-auto-action-types(3) version-1994(0)}
```

-- Identificateurs d'objet MS

id-cp-ms-connection, id-crt-ms-access-88, id-crt-ms-access-94, id-ot-ms, id-ot-ms-user, id-pt-retrieval-88,
id-pt-retrieval-94, id-pt-ms-submission

```
----
FROM MSObjectIdentifiers {joint-iso-itu-t mhs(6) ms(4) modules(0) object-identifiers(0)
                           version-1994(0)}
```

-- Codes des erreurs et des opérations abstraites de l'accès MS

err-attribute-error, err-auto-action-request-error, err-ms-extension-error,
err-delete-error, err-entry-class-error, err-fetch-restriction-error,
err-invalid-parameters-error, err-message-group-error, err-modify-error,
err-range-error, err-security-error, err-sequence-number-error, err-service-error,
err-register-ms-error, op-alert, op-delete, op-fetch, op-list, op-modify,
op-ms-message-submission, op-ms-probe-submission, op-register-ms, op-summarize

```
----
FROM MSAccessProtocol {joint-iso-itu-t mhs(6) protocols(0) modules(0)
                       ms-access-protocol(2) version-1994(0)}
```

-- Bornes inférieures du service abstrait MS

ub-attributes-supported, ub-attribute-values, ub-auto-action-errors, ub-auto-actions, ub-auto-registrations,
ub-default-registrations, ub-entry-classes, ub-error-reasons, ub-extensions, ub-group-depth,
ub-group-descriptor-length, ub-group-part-length, ub-matching-rules, ub-message-groups, ub-messages,
ub-modifications, ub-per-entry, ub-per-auto-action, ub-service-information-length, ub-summaries, ub-
supplementary-info-length,
ub-ua-registration-identifier-length, ub-ua-registrations, ub-restrictions

```
----
FROM MSUpperBounds {joint-iso-itu-t mhs(6) ms(4) modules(0) upper-bounds(4)
                    version-1994(0)}
```

-- Classe d'objets informationnels MATCHING-RULE

MATCHING-RULE

```
----
FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1) 2}
```

-- Opérations distantes

CONTRACT, CONNECTION-PACKAGE

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

emptyUnbind

```
FROM Remote-Operations-Useful-Definitions {joint-iso-ccitt remote-operations(4)
                                             useful-definitions(7) version1(0)};
```

-- Objets abstraits MS

```
ms MHS-OBJECT ::= {
  IS      {mts-user}
  RESPONDS {ms-access-contract-88 | ms-access-contract-94}
  ID      id-ot-ms }
```

```
ms-user MHS-OBJECT ::= {
  INITIATES {ms-access-contract-88 | ms-access-contract-94}
  ID        id-ot-ms-user }
```

-- Contrats

```
ms-access-contract-94 CONTRACT ::= {
    CONNECTION          ms-connect
    INITIATOR CONSUMER OF {retrieval | ms-submission | administration}
    ID                  id-crt-ms-access-94 }
```

```
ms-access-contract-88 CONTRACT ::= {
    CONNECTION          ms-connect -- toutes les extensions de 1994 sont omises --
    INITIATOR CONSUMER OF {retrieval-88 | submission | administration-88}
    ID                  id-crt-ms-access-88 }
```

-- Bloc de connexions

```
ms-connect CONNECTION-PACKAGE ::= {
    BIND      ms-bind
    UNBIND    ms-unbind
    ID        id-cp-ms-connection }
```

-- Accès MS

```
retrieval PORT ::= {
    CONSUMER INVOKES {summarize | list | fetch | delete | register-MS,
                    ... -- extension de 1994 --,
                    modify}
    SUPPLIER INVOKES {alert}
    ID                id-pt-retrieval-94 }
```

```
retrieval-88 PORT ::= {
    -- Toutes les extensions de 1994 relatives aux opérations abstraites sont absentes --
    CONSUMER INVOKES {summarize | list | fetch | delete | register-MS}
    SUPPLIER INVOKES {alert}
    ID                id-pt-retrieval-88 }
```

```
ms-submission PORT ::= {
    CONSUMER INVOKES {ms-message-submission | ms-probe-submission | ms-cancel-deferred-delivery}
    SUPPLIER INVOKES {ms-submission-control}
    ID                id-pt-ms-submission }
```

-- Classe d'objets informationnels ATTRIBUTE

```
ATTRIBUTE ::= CLASS {
    &id                AttributeType UNIQUE,
    &Type,
    &equalityMatch    MATCHING-RULE OPTIONAL,
    &substringsMatch  MATCHING-RULE OPTIONAL,
    &orderingMatch    MATCHING-RULE OPTIONAL,
    &numeration        ENUMERATED {single-valued(0), multi-valued(1)},
    -- extension de 1994 --
    &OtherMatches     MATCHING-RULE OPTIONAL }
```

```
WITH SYNTAX {
    WITH ATTRIBUTE-SYNTAX    &Type,
    [EQUALITY MATCHING-RULE &equalityMatch,]
    [SUBSTRINGS MATCHING-RULE &substringsMatch,]
    [ORDERING MATCHING-RULE &orderingMatch,]
    [OTHER MATCHING-RULES &OtherMatches,]
    NUMERATION              &numeration,
    ID                      &id }
```

```
Attribute ::= SEQUENCE {
    attribute-type    ATTRIBUTE.&id ({AttributeTable}),
    attribute-values  SEQUENCE SIZE (1.. ub-attribute-values) OF ATTRIBUTE.&Type
                    ({AttributeTable} {@attribute-type}) }
```

AttributeType ::= OBJECT IDENTIFIER

-- Classe d'objets informationnels AUTO-ACTION

AUTO-ACTION ::= CLASS {
 &id **AutoActionType UNIQUE,**
 &RegistrationParameter **OPTIONAL,**
 &Errors **AUTO-ACTION-ERROR OPTIONAL }**

WITH SYNTAX {
 [REGISTRATION PARAMETER IS &RegistrationParameter]
 [ERRORS &Errors]
 IDENTIFIED BY &id }

AutoActionType ::= OBJECT IDENTIFIER

AutoActionRegistration ::= SEQUENCE {
 auto-action-type **AUTO-ACTION.&id ({AutoActionTable}),**
 registration-identifier **[0] INTEGER (1..ub-per-auto-action) DEFAULT 1,**
 registration-parameter **[1] AUTO-ACTION.&RegistrationParameter ({AutoActionTable}**
 {@auto-action-type}) OPTIONAL }

-- Classe d'objets informationnels AUTO-ACTION-ERROR

AUTO-ACTION-ERROR ::= ABSTRACT-ERROR

AutoActionError ::= SET {
 error-code **[0] AUTO-ACTION-ERROR.&errorCode ({AutoActionErrorTable}),**
 error-parameter **[1] AUTO-ACTION-ERROR.&ParameterType ({AutoActionErrorTable} {@error-code})**
 OPTIONAL }

-- Classe d'objets informationnels MS-EXTENSION

MS-EXTENSION ::= TYPE-IDENTIFIER

MSExtensionItem ::= INSTANCE OF MS-EXTENSION

MSExtensions ::= SEQUENCE SIZE (1..ub-extensions) OF MSExtensionItem

-- Types de données communs relatifs au modèle informationnel

EntryClass ::= INTEGER {
 delivery **(0),**
 -- extensions de 1994 --
 submission **(1),**
 draft **(2),**
 stored-message **(3),**
 delivery-log **(4),**
 submission-log **(5),**
 message-log **(6),**
 auto-action-log **(7) } (0..ub-entry-classes)**

EntryType ::= INTEGER {
 delivered-message **(0),**
 delivered-report **(1),**
 returned-content **(2),**
 -- extensions de 1994 --
 submitted-message **(3),**
 submitted-probe **(4),**
 draft-message **(5),**
 auto-action-event **(6) }**

SequenceNumber ::= INTEGER (0..ub-messages)

RetrievalStatus ::= INTEGER {
 new **(0),**
 listed **(1),**
 processed **(2) }**

MessageGroupName ::= SEQUENCE SIZE (1..ub-group-depth) OF GroupNamePart

GroupNamePart ::= GeneralString (SIZE (1..ub-group-part-length))

-- Opération abstraite MS-bind

ms-bind ABSTRACT-OPERATION ::= {

 ARGUMENT MSBindArgument
 RESULT MSBindResult
 ERRORS {ms-bind-error} }

MSBindArgument ::= SET {

 initiator-name ORAddressAndOrDirectoryName,
 initiator-credentials [2] InitiatorCredentials,
 security-context [3] IMPLICIT SecurityContext OPTIONAL,
 fetch-restrictions [4] Restrictions OPTIONAL -- aucune restriction par défaut--,
 ms-configuration-request [5] BOOLEAN DEFAULT FALSE,
 -- extensions de 1994 --
 ua-registration-identifier [6] RegistrationIdentifier OPTIONAL,
 bind-extensions [7] MSExtensions OPTIONAL }

Restrictions ::= SET {

 allowed-content-types [0] SET SIZE (1..ub-content-types) OF OBJECT IDENTIFIER OPTIONAL
 -- aucune restriction par défaut --,
 allowed-EITs [1] MS-EITs OPTIONAL -- aucune restriction par défaut --,
 maximum-attribute-length [2] INTEGER OPTIONAL -- aucune restriction par défaut -- }

MS-EITs ::= SET SIZE (1..ub-encoded-information-types) OF MS-EIT

MS-EIT ::= OBJECT IDENTIFIER

RegistrationIdentifier ::= PrintableString (SIZE (1..ub-ua-registration-identifier-length))

MSBindResult ::= SET {

 responder-credentials [2] ResponderCredentials,
 available-auto-actions [3] SET SIZE (1..ub-auto-actions) OF AUTO-ACTION.&id
 ({AutoActionTable}) OPTIONAL,
 available-attribute-types [4] SET SIZE (1..ub-attributes-supported) OF ATTRIBUTE.&id
 ({AttributeTable}) OPTIONAL,
 alert-indication [5] BOOLEAN DEFAULT FALSE,
 content-types-supported [6] SET SIZE (1..ub-content-types) OF OBJECT IDENTIFIER OPTIONAL,
 -- extensions de 1994 --
 entry-classes-supported [7] SET SIZE (1..ub-entry-classes) OF EntryClass OPTIONAL,
 matching-rules-supported [8] SET SIZE (1..ub-matching-rules) OF OBJECT IDENTIFIER OPTIONAL,
 additional-capabilities [9] MSExtensions OPTIONAL,
 message-group-depth [10] INTEGER (1..ub-group-depth) OPTIONAL,
 auto-action-error-indication [11] AutoActionErrorIndication OPTIONAL,
 unsupported-extensions [12] SET SIZE (1..ub-extensions) OF OBJECT IDENTIFIER OPTIONAL,
 ua-registration-id-unknown [13] BOOLEAN DEFAULT FALSE,
 service-information [14] GeneralString (SIZE (1..ub-service-information-length)) OPTIONAL }

AutoActionErrorIndication ::= CHOICE {

 indication-only [0] NULL,
 auto-action-log-entry [1] SequenceNumber }

ms-bind-error ABSTRACT-ERROR ::= {

 PARAMETER CHOICE {
 unqualified-error BindProblem,
 -- extension de 1994 --
 qualified-error SET {
 bind-problem [0] BindProblem,
 supplementary-information [1] GeneralString (SIZE (1..ub-supplementary-info-length)) OPTIONAL,
 bind-extension-errors [2] SET SIZE (1..ub-extensions) OF OBJECT IDENTIFIER OPTIONAL } }

BindProblem ::= ENUMERATED {

 authentication-error (0),
 unacceptable-security-context (1),
 unable-to-establish-association (2),
 ... -- extension de 1994 --,
 bind-extension-problem (3) }

-- Opération abstraite MS Unbind

ms-unbind ABSTRACT-OPERATION ::= emptyUnbind

-- Types de données communs

```

Range ::= CHOICE {
    sequence-number-range      [0] NumberRange,
    creation-time-range        [1] TimeRange }

NumberRange ::= SEQUENCE {
    from      [0] SequenceNumber OPTIONAL -- pas de borne inférieure si cette composante est omise --,
    to        [1] SequenceNumber OPTIONAL -- pas de borne supérieure si cette composante est omise -- }

TimeRange ::= SEQUENCE {
    from      [0] CreationTime OPTIONAL -- pas de borne inférieure si cette composante est omise --,
    to        [1] CreationTime OPTIONAL -- pas de borne supérieure si cette composante est omise -- }

CreationTime ::= UTCTime

Filter ::= CHOICE {
    item      [0] FilterItem,
    and       [1] SET OF Filter,
    or        [2] SET OF Filter,
    not       [3] Filter }

FilterItem ::= CHOICE {
    equality          [0] AttributeValueAssertion,
    substrings       [1] SEQUENCE {
        type          ATTRIBUTE.&id ({AttributeTable}),
        strings       SEQUENCE OF CHOICE {
            initial    [0] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}),
            any         [1] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}),
            final      [2] ATTRIBUTE.&Type ({AttributeTable} {@substrings.type}) } },
    greater-or-equal [2] AttributeValueAssertion,
    less-or-equal    [3] AttributeValueAssertion,
    present           [4] ATTRIBUTE.&id ({AttributeTable}),
    approximate-match [5] AttributeValueAssertion,
                    -- extension de 1994 --
    other-match      [6] MatchingRuleAssertion }

MatchingRuleAssertion ::= SEQUENCE {
    matching-rule      [0] MATCHING-RULE.&id ({MatchingRuleTable}),
    attribute-type     [1] ATTRIBUTE.&id,
    match-value        [2] MATCHING-RULE.&AssertionType ({MatchingRuleTable} {@matching-rule}) }

AttributeValueAssertion ::= SEQUENCE {
    attribute-type     ATTRIBUTE.&id ({AttributeTable}),
    attribute-value    ATTRIBUTE.&Type ({AttributeTable} {@attribute-type}) }

Selector ::= SET {
    child-entries [0] BOOLEAN DEFAULT FALSE,
    range         [1] Range OPTIONAL -- pas de borne par défaut --,
    filter        [2] Filter OPTIONAL -- par défaut, toutes les entrées sont dans l'intervalle spécifié --,
    limit         [3] INTEGER (1..ub-messages) OPTIONAL,
    override      [4] OverrideRestrictions OPTIONAL -- par défaut, toutes les restrictions
                    -- de recherche en vigueur s'appliquent -- }

OverrideRestrictions ::= BIT STRING {
    override-content-types-restriction (0),
    override-EITs-restriction (1),
    override-attribute-length-restriction (2) (SIZE (1.. ub-restrictions))

EntryInformationSelection ::= SET SIZE (0..ub-per-entry) OF AttributeSelection

AttributeSelection ::= SET {
    type          ATTRIBUTE.&id ({AttributeTable}),
    from          [0] INTEGER (1..ub-attribute-values) OPTIONAL -- utilisé si le type est multi-valué --,
    count         [1] INTEGER (1..ub-attribute-values) OPTIONAL -- utilisé si le type est multi-valué -- }

EntryInformation ::= SEQUENCE {
    sequence-number      SequenceNumber,
    attributes           SET SIZE (1..ub-per-entry) OF Attribute OPTIONAL,
                    -- extension de 1994 --
    value-count-exceeded [0] SET SIZE (1..ub-per-entry) OF AttributeValueCount OPTIONAL }

```

```
AttributeValueCount ::= SEQUENCE {
    type      [0] ATTRIBUTE.&id ({AttributeTable}),
    total     [1] INTEGER }
```

```
MSSubmissionOptions ::= SET {
    object-entry-class      [0] EntryClass (submission|submission-log|draft) OPTIONAL,
    disable-auto-modify     [1] BOOLEAN DEFAULT FALSE,
    add-message-group-names [2] SET SIZE (1..ub-message-groups) OF MessageGroupName OPTIONAL,
    ms-submission-extensions [3] MSExtensions OPTIONAL }
```

```
CommonSubmissionResults ::= SET {
    created-entry           [0] SequenceNumber OPTIONAL,
    auto-action-error-indication [1] AutoActionErrorIndication OPTIONAL,
    ms-submission-result-extensions [2] MSExtensions OPTIONAL }
```

-- Opérations abstraites de l'accès d'extraction

```
summarize ABSTRACT-OPERATION ::= {
    ARGUMENT SummarizeArgument
    RESULT    SummarizeResult
    ERRORS    {attribute-error | invalid-parameters-error | range-error |
               security-error | sequence-number-error | service-error,
               ... -- extensions de 1994 --,
               entry-class-error | ms-extension-error}
    CODE      op-summarize }
```

```
SummarizeArgument ::= SET {
    entry-class      [0] EntryClass DEFAULT delivery,
    selector         [1] Selector,
    summary-requests [2] SEQUENCE SIZE (1..ub-summaries) OF ATTRIBUTE.&id ({AttributeTable})
                       OPTIONAL -- absent si aucune récapitulation n'est demandée --,
                       -- extension de 1994 --
    summarize-extensions [3] MSExtensions OPTIONAL }
```

```
SummarizeResult ::= SET {
    next           [0] SequenceNumber OPTIONAL,
    count          [1] INTEGER (0..ub-messages) -- décompte des entrées sélectionnées --,
    span           [2] Span OPTIONAL -- portée des entrées sélectionnées --,
                       -- omis si le décompte vaut zéro --,
    summaries      [3] SEQUENCE SIZE (1..ub-summaries) OF Summary OPTIONAL,
                       -- extension de 1994 --
    summarize-result-extensions [4] MSExtensions OPTIONAL }
```

```
Span ::= SEQUENCE {
    lowest [0] SequenceNumber,
    highest [1] SequenceNumber }
```

```
Summary ::= SET {
    absent [0] INTEGER (1..ub-messages) OPTIONAL -- décompte des entrées lorsque l'attribut est absent --,
    present [1] SET SIZE (1..ub-attribute-values) OF -- un pour chaque valeur d'attribut présente --
                SEQUENCE {
                    type ATTRIBUTE.&id ({AttributeTable}),
                    value ATTRIBUTE.&Type ({AttributeTable}) {@.type}),
                    count INTEGER (1..ub-messages) } OPTIONAL }
```

--

```
list ABSTRACT-OPERATION ::= {
    ARGUMENT ListArgument
    RESULT    ListResult
    ERRORS    {attribute-error | invalid-parameters-error | range-error |
               security-error | sequence-number-error | service-error,
               ... -- extensions de 1994 --,
               entry-class-error | ms-extension-error}
    CODE      op-list }
```

```

ListArgument ::= SET {
    entry-class          [0] EntryClass DEFAULT delivery,
    selector             [1] Selector,
    requested-attributes [3] EntryInformationSelection OPTIONAL,
                        -- extension de 1994 --
    list-extensions      [4] MSExtensions OPTIONAL }

ListResult ::= SET {
    next                [0] SequenceNumber OPTIONAL,
    requested           [1] SEQUENCE SIZE (1..ub-messages) OF EntryInformation OPTIONAL
                        -- omis si aucune information n'est trouvée --,
                        -- extension de 1994 --
    list-result-extensions [2] MSExtensions OPTIONAL }

--

fetch ABSTRACT-OPERATION ::= {
    ARGUMENT    FetchArgument
    RESULT      FetchResult
    ERRORS      {attribute-error | fetch-restriction-error | invalid-parameters-error | range-error |
                security-error | sequence-number-error | service-error,
                ... -- extension de 1994 --,
                entry-class-error | ms-extension-error}
    CODE        op-fetch }

FetchArgument ::= SET {
    entry-class          [0] EntryClass DEFAULT delivery,
    item                CHOICE {
        search          [1] Selector,
        precise         [2] SequenceNumber },
    requested-attributes [3] EntryInformationSelection OPTIONAL,
                        -- extensions de 1994 --
    fetch-extensions    [4] MSExtensions OPTIONAL }

FetchResult ::= SET {
    entry-information    [0] EntryInformation OPTIONAL -- si une entrée a été sélectionnée --,
    list                [1] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
    next                [2] SequenceNumber OPTIONAL,
                        -- extension de 1994 --
    fetch-result-extensions [3] MSExtensions OPTIONAL }

--

delete ABSTRACT-OPERATION ::= {
    ARGUMENT    DeleteArgument
    RESULT      DeleteResult
    ERRORS      {delete-error | invalid-parameters-error | range-error | security-error |
                sequence-number-error | service-error,
                ... -- extensions de 1994 --,
                entry-class-error | ms-extension-error}
    CODE        op-delete }

DeleteArgument ::= SET {
    entry-class          [0] EntryClass DEFAULT delivery,
    items                CHOICE {
        selector        [1] Selector,
        sequence-numbers [2] SET SIZE (1..ub-messages) OF SequenceNumber },
                        -- extension de 1994 --
    delete-extensions    [3] MSExtensions OPTIONAL }

DeleteResult ::= CHOICE {
    delete-result-88     NULL,
                        -- extension de 1994 --
    delete-result-94     SET {
        entries-deleted [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
        delete-result-extensions [1] MSExtensions OPTIONAL } }

```

--

register-MS ABSTRACT-OPERATION ::= {

ARGUMENT	Register-MSArgument
RESULT	Register-MSResult
ERRORS	{attribute-error auto-action-request-error invalid-parameters-error security-error service-error old-credentials-incorrectly-specified new-credentials-unacceptable, ... -- extensions de 1994 --, message-group-error ms-extension-error register-ms-error}
CODE	op-register-ms }

Register-MSArgument ::= SET {

auto-action-registrations	[0] SET SIZE (1..ub-auto-registrations) OF AutoActionRegistration OPTIONAL,
auto-action-deregistrations	[1] SET SIZE (1..ub-auto-registrations) OF AutoActionDeregistration OPTIONAL,
list-attribute-defaults	[2] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id (AttributeTable) OPTIONAL,
fetch-attribute-defaults	[3] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id (AttributeTable)OPTIONAL,
change-credentials	[4] SEQUENCE {
old-credentials	[0] Credentials,
new-credentials	[1] Credentials} OPTIONAL,
user-security-labels	[5] SET SIZE (1..ub-labels-and-redirections) OF SecurityLabel OPTIONAL, -- extensions de 1994 --
ua-registrations	[6] SET SIZE (1..ub-ua-registrations) OF UARegistration OPTIONAL,
submission-defaults	[7] MSSubmissionOptions OPTIONAL,
message-group-registrations	[8] MessageGroupRegistrations OPTIONAL,
registration-status-request	[9] RegistrationTypes OPTIONAL,
register-ms-extensions	[10] MSExtensions OPTIONAL }

AutoActionDeregistration ::= SEQUENCE {

auto-action-type	AUTO-ACTION.&id (AutoActionTable),
registration-identifier	[0] INTEGER (1..ub-per-auto-action) DEFAULT 1 }

UARegistration ::= SET {

ua-registration-identifier	[0] RegistrationIdentifier,
ua-list-attribute-defaults	[1] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id (AttributeTable) OPTIONAL,
ua-fetch-attribute-defaults	[2] SET SIZE (0..ub-default-registrations) OF ATTRIBUTE.&id (AttributeTable) OPTIONAL,
ua-submission-defaults	[3] MSSubmissionOptions OPTIONAL,
content-specific-defaults	[4] MSExtensions OPTIONAL }

MessageGroupRegistrations ::= SEQUENCE SIZE (1..ub-default-registrations) OF CHOICE {

register-group	[0] MessageGroupNameAndDescriptor,
deregister-group	[1] MessageGroupName,
change-descriptors	[2] MessageGroupNameAndDescriptor }

MessageGroupNameAndDescriptor ::= SET {

message-group-name	[0] MessageGroupName,
message-group-descriptor	[1] GeneralString (SIZE (1..ub-group-descriptor-length)) OPTIONAL }

RegistrationTypes ::= SET {

registrations	[0] BIT STRING {
auto-action-registrations	(0),
list-attribute-defaults	(1),
fetch-attribute-defaults	(2),
ua-registrations	(3),
submission-defaults	(4),
message-group-registrations	(5) } OPTIONAL,
extended-registrations	[1] SET OF MS-EXTENSION.&id OPTIONAL,
restrict-message-groups	[2] MessageGroupsRestriction OPTIONAL }

MessageGroupsRestriction ::= SET {

parent-group	[0] MessageGroupName OPTIONAL,
immediate-descendants-only	[1] BOOLEAN DEFAULT TRUE,
omit-descriptors	[2] BOOLEAN DEFAULT TRUE }

```

Register-MSResult ::= CHOICE {
    no-status-information          NULL,
                                -- extension de 1994 --
    registered-information SET {
        auto-action-registrations [0] SET SIZE (1..ub-auto-registrations) OF AutoActionRegistration
                                OPTIONAL,
        list-attribute-defaults   [1] SET SIZE (1..ub-default-registrations) OF ATTRIBUTE.&id
                                ({{AttributeTable}}) OPTIONAL,
        fetch-attribute-defaults [2] SET SIZE (1..ub-default-registrations) OF ATTRIBUTE.&id
                                ({{AttributeTable}}) OPTIONAL,
        ua-registrations           [3] SET SIZE (1..ub-ua-registrations) OF UARegistration OPTIONAL,
        submission-defaults       [4] MSSubmissionOptions OPTIONAL,
        message-group-registrations [5] SET SIZE (1..ub-message-groups) OF
                                MessageGroupNameAndDescriptor OPTIONAL,
        register-ms-result-extensions [6] MSExtensions OPTIONAL } }

```

--

```

alert ABSTRACT-OPERATION ::= {
    ARGUMENT AlertArgument
    RESULT   AlertResult
    ERRORS   {security-error}
    CODE     op-alert }

```

```

AlertArgument ::= SET {
    alert-registration-identifier [0] INTEGER (1..ub-auto-actions),
    new-entry                    [2] EntryInformation OPTIONAL }

```

```

AlertResult ::= NULL

```

--

```

modify ABSTRACT-OPERATION ::= {
    ARGUMENT   ModifyArgument
    RESULT     ModifyResult
    ERRORS     {attribute-error | invalid-parameters-error | security-error | sequence-number-error |
                service-error | modify-error | message-group-error | entry-class-error |
                ms-extension-error,
                ... -- Réserve pour de futures extensions --}
    CODE       op-modify }

```

```

ModifyArgument ::= SET {
    entry-class      [0] EntryClass DEFAULT delivery,
    entries          CHOICE {
        selector     [1] Selector,
        specific-entries [2] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber },
    modifications    [3] SEQUENCE SIZE (1..ub-modifications) OF EntryModification,
    modify-extensions [4] MSExtensions OPTIONAL }

```

```

EntryModification ::= SET {
    strict           [0] BOOLEAN DEFAULT FALSE,
    modification    CHOICE {
        add-attribute [1] Attribute,
        remove-attribute [2] ATTRIBUTE.&id ({{AttributeTable}}),
        add-values     [3] OrderedAttribute,
        remove-values  [4] OrderedAttribute } }

```

```

OrderedAttribute ::= SEQUENCE {
    attribute-type  ATTRIBUTE.&id ({{AttributeTable}}),
    attribute-values SEQUENCE SIZE (1..ub-attribute-values) OF SEQUENCE {
        -- il faut spécifier au moins une valeur --
        value       [0] ATTRIBUTE.&Type ({{AttributeTable}} {@attribute-type}) OPTIONAL,
        position    [1] INTEGER (1..ub-attribute-values) OPTIONAL } }

```

```

ModifyResult ::= SET {
    entries-modified [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
    modify-result-extensions [1] MSExtensions OPTIONAL }

```

-- Opérations abstraites de l'accès de dépôt MS

```
ms-message-submission ABSTRACT-OPERATION ::= {
  ARGUMENT      MSMessageSubmissionArgument
  RESULT        MSMessageSubmissionResult
  ERRORS        {submission-control-violated | element-of-service-not-subscribed | originator-invalid |
  recipient-improperly-specified | inconsistent-request | security-error |
  unsupported-critical-function | remote-bind-error,
  ... -- extensions de 1994 --,
  ms-extension-error | message-group-error | entry-class-error | service-error}
  CODE          op-ms-message-submission }
```

```
MSMessageSubmissionArgument ::= SEQUENCE {
  COMPONENTS OF MessageSubmissionArgument -- ce type importé est étiqueté implicitement --,
  -- extension de 1994 --
  submission-options [4] MSSubmissionOptions OPTIONAL }
```

```
forwarding-request EXTENSION ::= {
  SequenceNumber,
  IDENTIFIED BY standard-extension:36 }
```

```
MSMessageSubmissionResult ::= CHOICE {
  mts-result SET {
    COMPONENTS OF MessageSubmissionResult -- ce type importé est étiqueté implicitement --,
    -- extension de 1994 --
    ms-message-result [4] CommonSubmissionResults OPTIONAL },
    -- extension de 1994 --
  store-draft-result [4] CommonSubmissionResults }
```

--

```
ms-probe-submission ABSTRACT-OPERATION ::= {
  ARGUMENT      MSProbeSubmissionArgument
  RESULT        MSProbeSubmissionResult
  ERRORS        {submission-control-violated | element-of-service-not-subscribed | originator-invalid |
  recipient-improperly-specified | inconsistent-request | security-error |
  unsupported-critical-function | remote-bind-error,
  ... -- extensions de 1994 --,
  ms-extension-error | message-group-error | entry-class-error | service-error}
  CODE          op-ms-probe-submission }
```

```
MSProbeSubmissionArgument ::= SET {
  COMPONENTS OF ProbeSubmissionArgument -- ce type importé est étiqueté implicitement --,
  -- extension de 1994 --
  submission-options [4] MSSubmissionOptions OPTIONAL }
```

```
MSProbeSubmissionResult ::= SET {
  COMPONENTS OF ProbeSubmissionResult -- ce type importé est étiqueté implicitement --,
  -- extension de 1994 --
  ms-probe-result [4] CommonSubmissionResults OPTIONAL }
```

```
ms-cancel-deferred-delivery ABSTRACT-OPERATION ::= cancel-deferred-delivery
```

```
ms-submission-control ABSTRACT-OPERATION ::= submission-control
```

-- Erreurs abstraites

```
attribute-error ABSTRACT-ERROR ::= {
  PARAMETER SET {
    problems [0] SET SIZE (1.. ub-per-entry) OF SET {
      problem [0] AttributeProblem,
      type [1] ATTRIBUTE.&id ({AttributeTable}),
      value [2] ATTRIBUTE.&Type ({AttributeTable} {@.type}) OPTIONAL } }
  CODE err-attribute-error }
```

```

AttributeProblem ::= INTEGER {
    invalid-attribute-value      (0),
    unavailable-attribute-type   (1),
    inappropriate-matching      (2),
    attribute-type-not-subscribed (3),
    inappropriate-for-operation  (4),
    -- extensions de 1994 --
    inappropriate-modification   (5),
    single-valued-attribute      (6) } (0..ub-error-reasons)

--

auto-action-request-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problems      [0] SET SIZE (1..ub-auto-registrations) OF SET {
            problem    [0] AutoActionRequestProblem,
            type        [1] AUTO-ACTION.&id ({AutoActionTable}) } }
    CODE      err-auto-action-request-error }

AutoActionRequestProblem ::= INTEGER {
    unavailable-auto-action-type (0),
    auto-action-type-not-subscribed (1),
    -- extension de 1994 --
    not-willing-to-perform      (2) } (0..ub-error-reasons)

--

delete-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problems      [0] SET SIZE (1..ub-messages) OF SET {
            problem    [0] DeleteProblem,
            sequence-number [1] SequenceNumber },
        -- extension de 1994 --
        entries-deleted [1] SET SIZE (1..ub-messages) OF SequenceNumber OPTIONAL }
    CODE      err-delete-error }

DeleteProblem ::= INTEGER {
    child-entry-specified (0),
    delete-restriction-problem (1),
    -- extensions de 1994 --
    new-entry-specified (2),
    entry-class-restriction (3),
    stored-message-exists (4) } (0..ub-error-reasons)

--

fetch-restriction-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problems      [0] SET SIZE (1..ub-default-registrations) OF SET {
            problem    [3] FetchRestrictionProblem,
            restriction CHOICE {
                content-type [0] OBJECT IDENTIFIER,
                eit           [1] MS-EITs,
                attribute-length [2] INTEGER } } }
    CODE      err-fetch-restriction-error }

FetchRestrictionProblem ::= INTEGER {
    content-type-problem (1),
    eit-problem (2),
    maximum-length-problem (3) } (0..ub-error-reasons)

--

invalid-parameters-error ABSTRACT-ERROR ::= {
    PARAMETER NULL
    CODE      err-invalid-parameters-error }

--

```

```

range-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problem    [0] RangeProblem }
    CODE    err-range-error }

RangeProblem ::= INTEGER {
    reversed      (0) } (0..ub-error-reasons)

--

sequence-number-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problems [1] SET SIZE (1..ub-messages) OF SET {
            problem    [0] SequenceNumberProblem,
            sequence-number [1] SequenceNumber } }
    CODE    err-sequence-number-error }

SequenceNumberProblem ::= INTEGER {
    no-such-entry    (0) } (0..ub-error-reasons)

--

service-error ABSTRACT-ERROR ::= {
    PARAMETER    ServiceErrorParameter
    CODE        err-service-error }

ServiceErrorParameter ::= SET {
    problem                [0] ServiceProblem,
                        -- extension de 1994 --
    supplementary-information [1] GeneralString (SIZE (1..ub-supplementary-info-length)) OPTIONAL }

ServiceProblem ::= INTEGER {
    busy                (0),
    unavailable         (1),
    unwilling-to-perform (2) } (0..ub-error-reasons)

--

message-group-error ABSTRACT-ERROR ::= {
    PARAMETER    MessageGroupErrorParameter
    CODE        err-message-group-error }

MessageGroupErrorParameter ::= SET {
    problem    [0] MessageGroupProblem,
    name      [1] MessageGroupName }

MessageGroupProblem ::= INTEGER {
    name-not-registered    (0),
    name-already-registered (1),
    parent-not-registered  (2),
    group-not-empty        (3),
    name-in-use            (4),
    child-group-registered (5),
    group-depth-exceeded   (6) } (0..ub-error-reasons)

--

ms-extension-error ABSTRACT-ERROR ::= {
    PARAMETER    MSExtensionErrorParameter
    CODE        err-ms-extension-error }

MSExtensionErrorParameter ::= CHOICE {
    ms-extension-problem    [0] MSExtensionItem,
    unknown-ms-extension    [1] OBJECT IDENTIFIER }

--

register-ms-error ABSTRACT-ERROR ::= {
    PARAMETER SET {
        problem            [0] RegistrationProblem,
        registration-type  [1] RegistrationTypes }
    CODE    err-register-ms-error }

```

```

RegistrationProblem ::= ENUMERATED {
    registration-not-supported      (0),
    registration-improperly-specified (1),
    registration-limit-exceeded     (2),
    ... -- Réserve pour de futures extensions -- }

```

--

```

modify-error ABSTRACT-ERROR ::= {
    PARAMETER  ModifyErrorParameter
    CODE      err-modify-error }

```

```

ModifyErrorParameter ::= SET {
    entries-modified      [0] SEQUENCE SIZE (1..ub-messages) OF SequenceNumber OPTIONAL,
    failing-entry         [1] SequenceNumber,
    modification-number  [2] INTEGER,
    problem               [3] ModifyProblem }

```

```

ModifyProblem ::= INTEGER {
    attribute-not-present      (0),
    value-not-present         (1),
    attribute-or-value-already-exists (2),
    invalid-position          (3),
    modify-restriction-problem (4) } (0..ub-error-reasons)

```

--

```

entry-class-error ABSTRACT-ERROR ::= {
    PARAMETER  EntryClassErrorParameter
    CODE      err-entry-class-error }

```

```

EntryClassErrorParameter ::= SET {
    entry-class [0] EntryClass,
    problem    [1] BIT STRING {
        unsupported-entry-class (0),
        entry-class-not-subscribed (1),
        inappropriate-entry-class (2) } }

```

END -- fin du service abstrait MS

Annexe C

Définition formelle des types d'attributs généraux

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

La présente annexe, qui complète la section 3, définit de façon formelle les types d'attributs généraux applicables à toutes les formes de messagerie. Elle utilise la notation ASN.1 et la classe d'objets informationnels ATTRIBUTE.

```

-----
MSGGeneralAttributeTypes {joint-iso-itu-t mhs(6) ms(4) modules(0) general-attribute-types(2)
                           version-1994(0)}
DEFINITIONS ::=
BEGIN
-- Prologue

IMPORTS

-- Classe d'objets informationnels ATTRIBUTE

ATTRIBUTE,

-- Types de données du service abstrait MS

AutoActionError, AutoActionType, CreationTime, EntryClassErrorParameter, EntryType, MessageGroupName,
MessageGroupErrorParameter, MS-EIT, MSExtensionErrorParameter, RetrievalStatus, SequenceNumber,
ServiceErrorParameter
----
    FROM MSAbstractService {joint-iso-itu-t mhs(6) ms(4) modules(0) abstract-service(1)
                           version-1994(0)}

-- Identificateurs d'objet des types d'attributs généraux

id-att-ac-correlated-report-list, id-att-ac-report-summary, id-att-ac-uncorrelated-report-list,
id-att-auto-action-error, id-att-auto-action-registration-identifier, id-att-auto-action-subject-entry,
id-att-auto-action-type, id-att-child-sequence-numbers, id-att-content,
id-att-content-confidentiality-algorithm-identifier, id-att-content-correlator, id-att-content-identifier,
id-att-content-integrity-check, id-att-content-length, id-att-content-returned, id-att-content-type,
id-att-conversion-with-loss-prohibited, id-att-converted-EITs, id-att-creation-time,
id-att-deferred-delivery-cancellation-time, id-att-deferred-delivery-time, id-att-deletion-time, id-att-delivered-EITs,
id-att-delivery-flags, id-att-dl-expansion-history, id-att-dl-expansion-prohibited, id-att-entry-type,
id-att-internal-trace-information, id-att-latest-delivery-time, id-att-marked-for-deletion,
id-att-message-delivery-envelope, id-att-message-delivery-time, id-att-message-group-name,
id-att-message-identifier, id-att-message-notes, id-att-message-origin-authentication-check,
id-att-message-security-label, id-att-message-submission-envelope, id-att-message-submission-time,
id-att-message-token, id-att-ms-originated, id-att-ms-submission-error, id-att-original-EITs,
id-att-originally-intended-recipient-name, id-att-originating-MTA-certificate, id-att-originator-certificate,
id-att-originator-name, id-att-originator-report-request, id-att-originator-return-address,
id-att-other-recipient-names, id-att-parent-sequence-number, id-att-per-message-indicators,
id-att-per-recipient-message-submission-fields, id-att-per-recipient-probe-submission-fields,
id-att-per-recipient-report-delivery-fields, id-att-priority, id-att-probe-origin-authentication-check,
id-att-probe-submission-envelope, id-att-proof-of-delivery-request, id-att-proof-of-submission,
id-att-recipient-names, id-att-recipient-reassignment-prohibited, id-att-redirect-history,
id-att-report-delivery-envelope, id-att-reporting-DL-name, id-att-reporting-MTA-certificate,
id-att-report-origin-authentication-check, id-att-retrieval-status, id-att-security-classification,
id-att-sequence-number, id-att-storage-period, id-att-storage-time, id-att-subject-submission-identifier,
id-att-this-recipient-name, id-att-trace-information
----
    FROM MSObjectIdentifiers {joint-iso-itu-t mhs(6) ms(4) modules(0) object-identifiers(0)
                              version-1994(0)}

```

-- Règles de criblage de la mémoire de messages

bitStringMatch, contentCorrelatorMatch, contentIdentifierMatch, mSSingleSubstringListElementsMatch, mSSingleSubstringListMatch, mSSingleSubstringMatch, mSSubstringsMatch, mSStringCaseSensitiveMatch, mSStringListElementsMatch, mSStringListMatch, mSStringMatch, mSStringOrderingMatch, mTSIdentifierMatch, oRAddressElementsMatch, oRAddressMatch, oRAddressSubstringElementsMatch, oRNameElementsMatch, oRNameMatch, oRNameSingleElementMatch, oRNameSubstringElementsMatch, redirectionOrDLExpansionElementsMatch, redirectionOrDLExpansionMatch, redirectionOrDLExpansionSubstringElementsMatch, redirectionReasonMatch, valueCountMatch

FROM MSMatchingRules {joint-iso-itu-t mhs(6) ms(4) modules(0) general-matching-rules(5)}

-- Bornes supérieures du service abstrait MS

ub-entry-types, ub-message-notes-length

FROM MSUpperBounds {joint-iso-itu-t mhs(6) ms(4) modules(0) upper-bounds(4)
version-1994(0)}

-- Types de données du service abstrait MTS

Content, ContentCorrelator, ContentIdentifier, ContentIntegrityCheck, ContentLength, ConversionWithLossProhibited, DeferredDeliveryTime, DeliveryFlags, DLExpansion, DLExpansionProhibited, ImproperlySpecifiedRecipients, LatestDeliveryTime, MessageDeliveryEnvelope, MessageDeliveryTime, MessageOriginAuthenticationCheck, MessageSecurityLabel, MessageSubmissionEnvelope, MessageSubmissionTime, MessageToken, MTSIdentifier, OriginatingMTACertificate, OriginatorCertificate, OriginatorReportRequest, OriginatorReturnAddress, ORName, PerMessageIndicators, PerRecipientMessageSubmissionFields, PerRecipientProbeSubmissionFields, PerRecipientReportDeliveryFields, Priority, ProbeOriginAuthenticationCheck, ProbeSubmissionEnvelope, ProofOfDeliveryRequest, ProofOfSubmission, RecipientReassignmentProhibited, Redirection, ReportDeliveryEnvelope, ReportingDLName, ReportingMTACertificate, ReportOriginAuthenticationCheck, SecurityClassification, SecurityProblem, SubjectSubmissionIdentifier

FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3) modules(0) mts-abstract-service(1)
version-1994(0)}

-- Borne supérieure du service abstrait MTS

ub-recipients

FROM MTSUpperBounds {joint-iso-itu-t mhs(6) mts(3) modules(0) upper-bounds(3)}

-- Types de données du service abstrait MTA

InternalTraceInformationElement, TraceInformationElement

FROM MTAAbstractService {joint-iso-itu-t mhs(6) mts(3) modules(0) mta-abstract-service(2)
version-1994(0)}

-- Règles de criblage de l'annuaire

booleanMatch, integerMatch, integerOrderingMatch, uTCTimeMatch, uTCTimeOrderingMatch

FROM SelectedAttributeTypes {joint-iso-ccitt ds(5) modules(1) selectedAttributeTypes(5) 2}

objectIdentifierMatch

FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1) 2}

-- Types de données du service d'authentification

AlgorithmIdentifier

FROM AuthenticationFramework {joint-iso-ccitt ds(5) modules(1) authenticationFramework(7) 2};

-- Tableau des attributs

AttributeTable ATTRIBUTE ::= {
GeneralAttributes | ContentSpecificAttributes }

GeneralAttributes ATTRIBUTE ::= {
 ms-child-sequence-numbers | mt-content | mt-content-confidentiality-algorithm-identifier | mt-content-correlator |
 mt-content-identifier | mt-content-integrity-check | ms-content-length | ms-content-returned | mt-content-type |
 mt-conversion-with-loss-prohibited | mt-converted-EITs | ms-creation-time | ms-delivered-EITs | mt-delivery-flags |
 mt-dl-expansion-history | ms-entry-type | mt-message-delivery-envelope | mt-message-delivery-time |
 mt-message-identifier | mt-message-origin-authentication-check | mt-message-security-label |
 mt-message-submission-time | mt-message-token | mt-original-EITs | mt-originally-intended-recipient-name |
 mt-originator-certificate | mt-originator-name | mt-other-recipient-names | ms-parent-sequence-number |
 mt-per-recipient-report-delivery-fields | mt-priority | mt-proof-of-delivery-request | mt-redirect-history |
 mt-report-delivery-envelope | mt-reporting-DL-name | mt-reporting-MTA-certificate |
 mt-report-origin-authentication-check | ms-retrieval-status | mt-security-classification | ms-sequence-number |
 mt-subject-submission-identifier | mt-this-recipient-name,
 ... -- extensions de 1994 -- ,
 ms-ac-correlated-report-list | ms-ac-report-summary | ms-ac-uncorrelated-report-list | ms-auto-action-error |
 ms-auto-action-registration-identifier | ms-auto-action-subject-entry | ms-auto-action-type |
 ms-deferred-delivery-cancellation-time | mt-deferred-delivery-time | ms-deletion-time | mt-dl-expansion-prohibited |
 mt-internal-trace-information | mt-latest-delivery-time | ms-marked-for-deletion | ms-message-group-name |
 ms-message-notes | mt-message-submission-envelope | ms-originated | ms-submission-error |
 mt-originating-MTA-certificate | mt-originator-report-request | mt-originator-return-address |
 mt-per-message-indicators | mt-per-recipient-message-submission-fields | mt-per-recipient-probe-submission-fields |
 mt-probe-origin-authentication-check | mt-probe-submission-envelope | mt-proof-of-submission |
 ms-recipient-names | mt-recipient-reassignment-prohibited | ms-storage-period | ms-storage-time |
 mt-trace-information }

ContentSpecificAttributes ATTRIBUTE ::= { ... }

-- Types d'attribut

ms-ac-correlated-report-list ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX ReportLocation,
 NUMERATION multi-valued,
 ID id-att-ac-correlated-report-list }

ReportLocation ::= CHOICE {
 no-correlated-reports [0] NULL,
 location [1] SEQUENCE OF PerRecipientReport }

PerRecipientReport ::= SEQUENCE {
 report-entry [0] SequenceNumber,
 position [1] INTEGER (1..ub-recipients) DEFAULT 1 }

ms-ac-report-summary ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX ReportSummary,
 EQUALITY MATCHING-RULE integerMatch,
 ORDERING MATCHING-RULE integerOrderingMatch,
 NUMERATION multi-valued,
 ID id-att-ac-report-summary }

ReportSummary ::= ENUMERATED {
 no-report-requested (0) -- rapport de non-remise supprimé -- ,
 no-report-received (1) -- rapport de non-remise demandé -- ,
 report-outstanding (2) -- rapport de remise demandé -- ,
 delivery-cancelled (3),
 delivery-report-from-another-recipient (4),
 non-delivery-report-from-another-recipient (5),
 delivery-report-from-intended-recipient (6),
 non-delivery-report-from-intended-recipient (7) }

ms-ac-uncorrelated-report-list ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX PerRecipientReport,
 NUMERATION multi-valued,
 ID id-att-ac-uncorrelated-report-list }

ms-auto-action-error ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX AutoActionError,
 NUMERATION single-valued,
 ID id-att-auto-action-error }

ms-auto-action-registration-identifier ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX INTEGER,
EQUALITY MATCHING-RULE integerMatch,
ORDERING MATCHING-RULE integerOrderingMatch,
NUMERATION single-valued,
ID id-att-auto-action-registration-identifier }

ms-auto-action-subject-entry ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX SequenceNumber,
EQUALITY MATCHING-RULE integerMatch,
ORDERING MATCHING-RULE integerOrderingMatch,
NUMERATION single-valued,
ID id-att-auto-action-subject-entry }

ms-auto-action-type ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX AutoActionType,
EQUALITY MATCHING-RULE objectIdentifierMatch,
NUMERATION single-valued,
ID id-att-auto-action-type }

ms-child-sequence-numbers ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX SequenceNumber,
NUMERATION multi-valued,
ID id-att-child-sequence-numbers }

mt-content ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX Content,
NUMERATION single-valued,
ID id-att-content }

mt-content-confidentiality-algorithm-identifier ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX AlgorithmIdentifier,
NUMERATION single-valued,
ID id-att-content-confidentiality-algorithm-identifier }

mt-content-correlator ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX ContentCorrelator,
EQUALITY MATCHING-RULE contentCorrelatorMatch,
NUMERATION single-valued,
ID id-att-content-correlator }

mt-content-identifier ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX ContentIdentifier,
EQUALITY MATCHING-RULE contentIdentifierMatch,
NUMERATION single-valued,
ID id-att-content-identifier }

mt-content-integrity-check ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX ContentIntegrityCheck,
NUMERATION single-valued,
ID id-att-content-integrity-check }

ms-content-length ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX ContentLength,
ORDERING MATCHING-RULE integerOrderingMatch,
NUMERATION single-valued,
ID id-att-content-length }

ms-content-returned ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX BOOLEAN,
EQUALITY MATCHING-RULE booleanMatch,
NUMERATION single-valued,
ID id-att-content-returned }

mt-content-type ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER,
EQUALITY MATCHING-RULE objectIdentifierMatch,
NUMERATION single-valued,
ID id-att-content-type }

mt-conversion-with-loss-prohibited ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX ConversionWithLossProhibited,
EQUALITY MATCHING-RULE integerMatch,
NUMERATION single-valued,
ID id-att-conversion-with-loss-prohibited }

mt-converted-EITs ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX MS-EIT,
EQUALITY MATCHING-RULE objectIdentifierMatch,
NUMERATION multi-valued,
ID id-att-converted-EITs }

ms-creation-time ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX CreationTime,
EQUALITY MATCHING-RULE uTCTimeMatch,
ORDERING MATCHING-RULE uTCTimeOrderingMatch,
NUMERATION single-valued,
ID id-att-creation-time }

ms-deferred-delivery-cancellation-time ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DeferredDeliveryCancellationTime,
EQUALITY MATCHING-RULE uTCTimeMatch,
ORDERING MATCHING-RULE uTCTimeOrderingMatch,
NUMERATION single-valued,
ID id-att-deferred-delivery-cancellation-time }

DeferredDeliveryCancellationTime ::= UTCTime

mt-deferred-delivery-time ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DeferredDeliveryTime,
EQUALITY MATCHING-RULE uTCTimeMatch,
ORDERING MATCHING-RULE uTCTimeOrderingMatch,
NUMERATION single-valued,
ID id-att-deferred-delivery-time }

ms-deletion-time ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DeletionTime,
EQUALITY MATCHING-RULE uTCTimeMatch,
ORDERING MATCHING-RULE uTCTimeOrderingMatch,
NUMERATION single-valued,
ID id-att-deletion-time }

DeletionTime ::= UTCTime

ms-delivered-EITs ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX MS-EIT,
EQUALITY MATCHING-RULE objectIdentifierMatch,
NUMERATION multi-valued,
ID id-att-delivered-EITs }

mt-delivery-flags ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DeliveryFlags,
EQUALITY MATCHING-RULE bitStringMatch,
NUMERATION single-valued,
ID id-att-delivery-flags }

mt-dl-expansion-history ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DLExpansion,
OTHER MATCHING-RULES {redirectionOrDLExpansionMatch | redirectionOrDLExpansionElementsMatch | redirectionOrDLExpansionSubstringElementsMatch},
NUMERATION multi-valued,
ID id-att-dl-expansion-history }

mt-dl-expansion-prohibited ATTRIBUTE ::= {
WITH ATTRIBUTE-SYNTAX DLExpansionProhibited,
EQUALITY MATCHING-RULE integerMatch,
NUMERATION single-valued,
ID id-att-dl-expansion-prohibited }

ms-entry-type ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	EntryType,
EQUALITY MATCHING-RULE	integerMatch,
ORDERING MATCHING-RULE	integerOrderingMatch, -- règle non définie dans les contextes d'application 1988 --
NUMERATION	single-valued,
ID	id-att-entry-type }
mt-internal-trace-information ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	InternalTraceInformationElement,
NUMERATION	multi-valued,
ID	id-att-internal-trace-information }
mt-latest-delivery-time ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	LatestDeliveryTime,
EQUALITY MATCHING-RULE	uTCTimeMatch,
ORDERING MATCHING-RULE	uTCTimeOrderingMatch,
NUMERATION	single-valued,
ID	id-att-latest-delivery-time }
ms-marked-for-deletion ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	NULL,
NUMERATION	single-valued,
ID	id-att-marked-for-deletion }
mt-message-delivery-envelope ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageDeliveryEnvelope,
NUMERATION	single-valued,
ID	id-att-message-delivery-envelope }
mt-message-delivery-time ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageDeliveryTime,
EQUALITY MATCHING-RULE	uTCTimeMatch,
ORDERING MATCHING-RULE	uTCTimeOrderingMatch,
NUMERATION	single-valued,
ID	id-att-message-delivery-time }
ms-message-group-name ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageGroupName,
EQUALITY MATCHING-RULE	mSStringListMatch,
OTHER MATCHING-RULES	{mSSingleSubstringListMatch mSStringListElementsMatch mSSingleSubstringListElementsMatch valueCountMatch},
NUMERATION	multi-valued,
ID	id-att-message-group-name }
mt-message-identifier ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MTSIdentifier,
EQUALITY MATCHING-RULE	mTIdentifierMatch, -- règle non définie dans les contextes d'application 1988 --
NUMERATION	single-valued,
ID	id-att-message-identifier }
ms-message-notes ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	GeneralString (SIZE (1..ub-message-notes-length)),
EQUALITY MATCHING-RULE	mSStringMatch,
SUBSTRINGS MATCHING-RULE	mSSubstringsMatch,
NUMERATION	multi-valued,
ID	id-att-message-notes }
mt-message-origin-authentication-check ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageOriginAuthenticationCheck,
NUMERATION	single-valued,
ID	id-att-message-origin-authentication-check }
mt-message-security-label ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageSecurityLabel,
NUMERATION	single-valued,
ID	id-att-message-security-label }
mt-message-submission-envelope ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageSubmissionEnvelope,
NUMERATION	single-valued,
ID	id-att-message-submission-envelope }

mt-message-submission-time ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageSubmissionTime,
EQUALITY MATCHING-RULE	uTCTimeMatch,
ORDERING MATCHING-RULE	uTCTimeOrderingMatch,
NUMERATION	single-valued,
ID	id-att-message-submission-time }
mt-message-token ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MessageToken,
NUMERATION	single-valued,
ID	id-att-message-token }
ms-originated ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	NULL,
NUMERATION	single-valued,
ID	id-att-ms-originated }
ms-submission-error ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	SubmissionError,
NUMERATION	single-valued,
ID	id-att-ms-submission-error }
SubmissionError ::= CHOICE {	
submission-control-violated	[1] NULL,
originator-invalid	[2] NULL,
recipient-improperly-specified	[3] ImproperlySpecifiedRecipients,
element-of-service-not-subscribed	[4] NULL,
inconsistent-request	[11] NULL,
security-error	[12] SecurityProblem,
unsupported-critical-function	[13] NULL,
remote-bind-error	[15] NULL,
service-error	[27] ServiceErrorParameter,
message-group-error	[30] MessageGroupErrorParameter,
ms-extension-error	[31] MSExtensionErrorParameter,
entry-class-error	[34] EntryClassErrorParameter }
mt-original-EITs ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	MS-EIT,
EQUALITY MATCHING-RULE	objectIdentifierMatch,
NUMERATION	multi-valued,
ID	id-att-original-EITs }
mt-originally-intended-recipient-name ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ORName,
EQUALITY MATCHING-RULE	oRNameMatch,
OTHER MATCHING-RULES	{oRNameElementsMatch oRNameSubstringElementsMatch
	oRNameSingleElementMatch},
NUMERATION	single-valued,
ID	id-att-originally-intended-recipient-name }
mt-originating-MTA-certificate ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	OriginatingMTACertificate,
NUMERATION	single-valued,
ID	id-att-originating-MTA-certificate }
mt-originator-certificate ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	OriginatorCertificate,
NUMERATION	single-valued,
ID	id-att-originator-certificate }
mt-originator-name ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ORName,
EQUALITY MATCHING-RULE	oRNameMatch,
OTHER MATCHING-RULES	{oRNameElementsMatch oRNameSubstringElementsMatch
	oRNameSingleElementMatch},
NUMERATION	single-valued,
ID	id-att-originator-name }
mt-originator-report-request ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	OriginatorReportRequest,
NUMERATION	multi-valued,
ID	id-att-originator-report-request }

mt-originator-return-address ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	OriginatorReturnAddress,
NUMERATION	single-valued,
ID	id-att-originator-return-address }
mt-other-recipient-names ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ORName,
EQUALITY MATCHING-RULE	oRNameMatch,
OTHER MATCHING-RULES	{oRNameElementsMatch oRNameSubstringElementsMatch
	oRNameSingleElementMatch},
NUMERATION	multi-valued,
ID	id-att-other-recipient-names }
ms-parent-sequence-number ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	SequenceNumber,
EQUALITY MATCHING-RULE	integerMatch,
ORDERING MATCHING-RULE	integerOrderingMatch,
NUMERATION	single-valued,
ID	id-att-parent-sequence-number }
mt-per-message-indicators ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	PerMessageIndicators,
EQUALITY MATCHING-RULE	bitStringMatch,
NUMERATION	single-valued,
ID	id-att-per-message-indicators }
mt-per-recipient-message-submission-fields ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	PerRecipientMessageSubmissionFields,
NUMERATION	multi-valued,
ID	id-att-per-recipient-message-submission-fields }
mt-per-recipient-probe-submission-fields ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	PerRecipientProbeSubmissionFields,
NUMERATION	multi-valued,
ID	id-att-per-recipient-probe-submission-fields }
mt-per-recipient-report-delivery-fields ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	PerRecipientReportDeliveryFields,
NUMERATION	multi-valued,
ID	id-att-per-recipient-report-delivery-fields }
mt-priority ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	Priority,
EQUALITY MATCHING-RULE	integerMatch,
ORDERING MATCHING-RULE	integerOrderingMatch, -- règle non définie dans les contextes
	<i>d'application 1988 --</i>
NUMERATION	single-valued,
ID	id-att-priority }
mt-probe-origin-authentication-check ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ProbeOriginAuthenticationCheck,
NUMERATION	single-valued,
ID	id-att-probe-origin-authentication-check }
mt-probe-submission-envelope ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ProbeSubmissionEnvelope,
NUMERATION	single-valued,
ID	id-att-probe-submission-envelope }
mt-proof-of-delivery-request ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ProofOfDeliveryRequest,
EQUALITY MATCHING-RULE	integerMatch, -- règle non définie dans les contextes d'application 1988 --
NUMERATION	single-valued,
ID	id-att-proof-of-delivery-request }
mt-proof-of-submission ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ProofOfSubmission,
NUMERATION	single-valued,
ID	id-att-proof-of-submission }

ms-recipient-names ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ORName,
EQUALITY MATCHING-RULE	oRNameMatch,
OTHER MATCHING-RULES	{oRNameElementsMatch oRNameSubstringElementsMatch
	oRNameSingleElementMatch},
NUMERATION	multi-valued,
ID	id-att-recipient-names }
mt-recipient-reassignment-prohibited ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	RecipientReassignmentProhibited,
EQUALITY MATCHING-RULE	integerMatch,
NUMERATION	single-valued,
ID	id-att-recipient-reassignment-prohibited }
mt-redirectation-history ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	Redirection,
OTHER MATCHING-RULES	{redirectionOrDLExpansionMatch redirectionOrDLExpansionElementsMatch
	redirectionOrDLExpansionSubstringElementsMatch
	redirectionReasonMatch},
NUMERATION	multi-valued,
ID	id-att-redirectation-history }
mt-report-delivery-envelope ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ReportDeliveryEnvelope,
NUMERATION	single-valued,
ID	id-att-report-delivery-envelope }
mt-reporting-DL-name ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ReportingDLName,
EQUALITY MATCHING-RULE	oRNameMatch, -- règle non définie dans les contextes
	<i>d'application 1988 --</i>
OTHER MATCHING-RULES	{oRNameElementsMatch oRNameSubstringElementsMatch
	oRNameSingleElementMatch},
NUMERATION	single-valued,
ID	id-att-reporting-DL-name }
mt-reporting-MTA-certificate ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ReportingMTACertificate,
NUMERATION	single-valued,
ID	id-att-reporting-MTA-certificate }
mt-report-origin-authentication-check ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	ReportOriginAuthenticationCheck,
NUMERATION	single-valued,
ID	id-att-report-origin-authentication-check }
ms-retrieval-status ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	RetrievalStatus,
EQUALITY MATCHING-RULE	integerMatch,
NUMERATION	single-valued,
ID	id-att-retrieval-status }
mt-security-classification ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	SecurityClassification,
EQUALITY MATCHING-RULE	integerMatch,
NUMERATION	single-valued,
ID	id-att-security-classification }
ms-sequence-number ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	SequenceNumber,
EQUALITY MATCHING-RULE	integerMatch,
ORDERING MATCHING-RULE	integerOrderingMatch,
NUMERATION	single-valued,
ID	id-att-sequence-number }
ms-storage-time ATTRIBUTE ::= {	
WITH ATTRIBUTE-SYNTAX	StorageTime,
EQUALITY MATCHING-RULE	uTCTimeMatch,
ORDERING MATCHING-RULE	uTCTimeOrderingMatch,
NUMERATION	single-valued,
ID	id-att-storage-time }

StorageTime ::= UTCTime

ms-storage-period ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX StoragePeriod,
 EQUALITY MATCHING-RULE integerMatch,
 ORDERING MATCHING-RULE integerOrderingMatch,
 NUMERATION single-valued,
 ID id-att-storage-period }

StoragePeriod ::= INTEGER -- secondes --

mt-subject-submission-identifier ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX SubjectSubmissionIdentifier,
 EQUALITY MATCHING-RULE mTSIdentifierMatch, -- règle non définie dans les contextes
 d'application 1988 --
 NUMERATION single-valued,
 ID id-att-subject-submission-identifier }

mt-this-recipient-name ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX ORName,
 EQUALITY MATCHING-RULE oRNameMatch,
 OTHER MATCHING-RULES {oRNameElementsMatch | oRNameSubstringElementsMatch |
 oRNameSingleElementMatch},
 NUMERATION single-valued,
 ID id-att-this-recipient-name }

mt-trace-information ATTRIBUTE ::= {
 WITH ATTRIBUTE-SYNTAX TraceInformationElement,
 NUMERATION multi-valued,
 ID id-att-trace-information }

END -- fin des types d'attributs généraux MS

Annexe D

Définition formelle des règles de criblage générales

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

La présente annexe, qui complète la section 3, définit de façon formelle les règles de criblage générales applicables aux entrées des messages de tous les types de contenu.

```

-----
MSMatchingRules {joint-iso-itu-t mhs(6) ms(4) modules(0) general-matching-rules(5)}
DEFINITIONS ::=

BEGIN

-- Prologue

    -- Exporte tout

IMPORTS

    -- Classe d'objets informationnels MATCHING-RULE et règles de criblage de l'annuaire

MATCHING-RULE, objectIdentifierMatch
----
    FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1) 2}

bitStringMatch, booleanMatch, integerMatch, integerOrderingMatch, uTCTimeMatch, uTCTimeOrderingMatch
----
    FROM SelectedAttributeTypes {joint-iso-ccitt ds(5) modules(1) selectedAttributeTypes(5) 2}

    -- Identificateurs d'objet des règles de criblage

id-mr-content-correlator-match, id-mr-content-identifieur-match, id-mr-ms-single-substring-list-elements-match,
id-mr-ms-single-substring-list-match, id-mr-ms-single-substring-match, id-mr-ms-substrings-match,
id-mr-msstring-case-sensitive-match, id-mr-msstring-list-elements-match, id-mr-msstring-list-match,
id-mr-msstring-match, id-mr-msstring-ordering-match, id-mr-mts-identifieur-match,
id-mr-oraddress-elements-match, id-mr-oraddress-match, id-mr-oraddress-substring-elements-match,
id-mr-orname-elements-match, id-mr-orname-match, id-mr-orname-single-element-match,
id-mr-orname-substring-elements-match, id-mr-redirection-or-dl-expansion-elements-match,
id-mr-redirection-or-dl-expansion-match, id-mr-redirection-or-dl-expansion-substring-elements-match,
id-mr-redirection-reason-match, id-mr-value-count-match
----
    FROM MSObjectIdentifiers {joint-iso-itu-t mhs(6) ms(4) modules(0) object-identifiers(0)
                                version-1994(0)}

    -- Bornes supérieures MS

ub-attribute-values, ub-msstring-match
----
    FROM MSUpperBounds {joint-iso-itu-t mhs(6) ms(4) modules(0) upper-bounds(4)
                                version-1994(0)}

    -- Service abstrait MTS

ContentCorrelator, ContentIdentifier, MTSIdentifier, ORAddress, ORAddressAndOptionalDirectoryName,
ORName, RedirectionReason
----
    FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3) modules(0) mts-abstract-service(1)
                                version-1994(0)};

-- Ensemble d'objets informationnels Matching-rule-table

MatchingRuleTable MATCHING-RULE ::= {
    GeneralMatchingRules | ContentSpecificMatchingRules }

```

```

GeneralMatchingRules MATCHING-RULE ::= {
    bitStringMatch | booleanMatch | contentIdentifierMatch | integerMatch | integerOrderingMatch |
    mSStringCaseSensitiveMatch | objectIdentifierMatch | oRNameMatch | uTCTimeMatch | uTCTimeOrderingMatch,
    ... -- extensions de 1994 --,
    contentCorrelatorMatch | mSSingleSubstringMatch | mSStringCaseSensitiveMatch | mSStringListElementsMatch |
    mSStringListMatch | mSStringMatch | mSStringOrderingMatch | mSSingleSubstringListElementsMatch |
    mSSingleSubstringListMatch | mSSubstringsMatch | mTSIdentifierMatch | oRAddressElementsMatch |
    oRAddressMatch | oRAddressSubstringElementsMatch | oRNameElementsMatch | oRNameMatch |
    oRNameSingleElementMatch | oRNameSubstringElementsMatch | redirectionOrDLExpansionElementsMatch |
    redirectionOrDLExpansionMatch | redirectionOrDLExpansionSubstringElementsMatch |
    redirectionReasonMatch | valueCountMatch }

```

```

ContentSpecificMatchingRules MATCHING-RULE ::= { ... }

```

-- Syntaxe d'assertion MS-string

```

MSString {INTEGER : maxSize} ::= CHOICE {
    printable      PrintableString (SIZE (1..maxSize)),
    teletex        TeletexString (SIZE (1..maxSize)),
    general        GeneralString (SIZE (1..maxSize)),
    universal      UniversalString (SIZE (1..maxSize)),
    bmp            BMPString (SIZE (1..maxSize)) }

```

-- Règles de criblage basées sur les chaînes

```

mSStringMatch MATCHING-RULE ::= {
    SYNTAX MSString {ub-msstring-match}
    ID      id-mr-msstring-match }

```

```

mSStringOrderingMatch MATCHING-RULE ::= {
    SYNTAX MSString {ub-msstring-match}
    ID      id-mr-msstring-ordering-match }

```

```

mSSubstringsMatch MATCHING-RULE ::= {
    SYNTAX SubstringAssertion
    ID      id-mr-ms-substrings-match }

```

```

SubstringAssertion ::= SEQUENCE OF CHOICE {
    initial  [0] MSString {ub-msstring-match},
    any     [1] MSString {ub-msstring-match},
    final   [2] MSString {ub-msstring-match} }
-- au plus une composante initiale et une composante finale --

```

```

mSSingleSubstringMatch MATCHING-RULE ::= {
    SYNTAX MSString {ub-msstring-match}
    ID      id-mr-ms-single-substring-match }

```

```

mSStringCaseSensitiveMatch MATCHING-RULE ::= {
    SYNTAX MSString {ub-msstring-match}
    ID      id-mr-msstring-case-sensitive-match }

```

```

mSStringListMatch MATCHING-RULE ::= {
    SYNTAX SEQUENCE OF MSString {ub-msstring-match}
    ID      id-mr-msstring-list-match }

```

```

mSStringListElementsMatch MATCHING-RULE ::= {
    SYNTAX SEQUENCE OF MSString {ub-msstring-match}
    ID      id-mr-msstring-list-elements-match }

```

```

mSSingleSubstringListMatch MATCHING-RULE ::= {
    SYNTAX SEQUENCE OF MSString {ub-msstring-match}
    ID      id-mr-ms-single-substring-list-match }

```

```

mSSingleSubstringListElementsMatch MATCHING-RULE ::= {
    SYNTAX SEQUENCE OF MSString {ub-msstring-match}
    ID      id-mr-ms-single-substring-list-elements-match }

```

-- Règle de criblage basée sur la syntaxe

```

valueCountMatch MATCHING-RULE ::= {
    SYNTAX INTEGER (1..ub-attribute-values)
    ID      id-mr-value-count-match }

```

-- Règles de criblage pour les attributs complexes de la mémoire de messages

-- Règles de criblage d'adresse OR

```
oRAddressMatch MATCHING-RULE ::= {  
    SYNTAX ORAddress  
    ID      id-mr-oraddress-match }
```

```
oRAddressElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORAddress  
    ID      id-mr-oraddress-elements-match }
```

```
oRAddressSubstringElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORAddress  
    ID      id-mr-oraddress-substring-elements-match }
```

-- Règles de criblage de nom OR

```
oRNameMatch MATCHING-RULE ::= {  
    SYNTAX ORName  
    ID      id-mr-orname-match }
```

```
oRNameElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORName  
    ID      id-mr-orname-elements-match }
```

```
oRNameSubstringElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORName  
    ID      id-mr-orname-substring-elements-match }
```

```
oRNameSingleElementMatch MATCHING-RULE ::= {  
    SYNTAX MSSString {ub-msstring-match}  
    ID      id-mr-orname-single-element-match }
```

-- Règles de criblage concernant le réacheminement ou le développement de la liste de distribution

```
redirectionOrDLExpansionMatch MATCHING-RULE ::= {  
    SYNTAX ORAddressAndOptionalDirectoryName  
    ID      id-mr-redirection-or-dl-expansion-match }
```

```
redirectionOrDLExpansionElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORAddressAndOptionalDirectoryName  
    ID      id-mr-redirection-or-dl-expansion-elements-match }
```

```
redirectionOrDLExpansionSubstringElementsMatch MATCHING-RULE ::= {  
    SYNTAX ORAddressAndOptionalDirectoryName  
    ID      id-mr-redirection-or-dl-expansion-substring-elements-match }
```

```
redirectionReasonMatch MATCHING-RULE ::= {  
    SYNTAX RedirectionReason  
    ID      id-mr-redirection-reason-match }
```

-- Règle de criblage basée sur le type MTS-identifiant

```
mTSIdentifierMatch MATCHING-RULE ::= {  
    SYNTAX MTSIdentifier  
    ID      id-mr-mts-identifiant-match }
```

-- Règle de criblage basée sur le type content-correlator

```
contentCorrelatorMatch MATCHING-RULE ::= {  
    SYNTAX ContentCorrelator  
    ID      id-mr-content-correlator-match }
```

-- Règle de criblage basée sur le type content-identifiant

```
contentIdentifierMatch MATCHING-RULE ::= {  
    SYNTAX ContentIdentifier  
    ID      id-mr-content-identifiant-match }
```

END -- fin des règles de criblage MS

Annexe E

Définition formelle des types d'actions automatiques générales

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

La présente annexe, qui complète la section 3, définit de façon formelle les types d'actions automatiques générales applicables à toutes les formes de messagerie. Elle utilise la notation ASN.1 et la classe d'objets informationnels AUTO-ACTION (action automatique).

```

-----
MSGGeneralAutoActionTypes {joint-iso-itu-t mhs(6) ms(4) modules(0) general-auto-action-types(3)
                          version-1994(0)}
DEFINITIONS ::=
BEGIN
-- Prologue

-- Exporte tout

IMPORTS

-- Classes d'objets informationnels AUTO-ACTION et AUTO-ACTION-ERROR
AUTO-ACTION, AUTO-ACTION-ERROR,

-- Erreurs abstraites et types de données du service abstrait MS
EntryClass, EntryInformationSelection, EntryModification, Filter, message-group-error, modify-error,
service-error, SequenceNumber
----
FROM MSAbstractService {joint-iso-itu-t mhs(6) ms(4) modules(0) abstract-service(1)
                        version-1994(0)}

-- Identificateurs d'objets MS
id-aae-auto-alert-error, id-act-auto-alert, id-act-auto-correlate-reports, id-act-auto-delete, id-act-auto-modify
----
FROM MSObjectIdentifiers {joint-iso-itu-t mhs(6) ms(4) modules(0) object-identifiers(0)
                           version-1994(0)}

-- Bornes supérieures du service abstrait MS
ub-alert-addresses, ub-modifications, ub-supplementary-info-length
----
FROM MSUpperBounds {joint-iso-itu-t mhs(6) ms(4) modules(0) upper-bounds(4)
                    version-1994(0)}

-- Service abstrait MTS
security-error
----
FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3) modules(0) mts-abstract-service(1)
                         version-1994(0)};

-- Ensemble d'objets informationnels Auto-action-table
AutoActionTable AUTO-ACTION ::= {
    GeneralAutoActions | ContentSpecificAutoActions }

GeneralAutoActions AUTO-ACTION ::= {
    auto-alert,
    ... -- extensions de 1994 -- ,
    auto-modify | auto-correlate-reports | auto-delete }

ContentSpecificAutoActions AUTO-ACTION ::= { ... }

-- Ensemble d'objets informationnels Auto-action-error-table
AutoActionErrorTable AUTO-ACTION-ERROR ::= {
    GeneralAutoActionErrors | ContentSpecificAutoActionErrors }

```

GeneralAutoActionErrors AUTO-ACTION-ERROR ::= {
auto-alert-error | modify-error | service-error | security-error | message-group-error,
... -- Réservé pour de futures extensions -- }

ContentSpecificAutoActionErrors AUTO-ACTION-ERROR ::= { ... }

-- Types d'action automatique

auto-alert AUTO-ACTION ::= {
REGISTRATION PARAMETER IS AutoAlertRegistrationParameter
ERRORS {auto-alert-error}
IDENTIFIED BY id-act-auto-alert }

AutoAlertRegistrationParameter ::= SET {
filter [0] Filter OPTIONAL,
alert-destinations [1] SEQUENCE SIZE (1..ub-alert-addresses) OF AlertDestination OPTIONAL,
requested-attributes [2] EntryInformationSelection OPTIONAL,
-- extension de 1994 --
suppress-alert-destinations [3] BOOLEAN DEFAULT TRUE }

AlertDestination ::= SEQUENCE {
alert-address EXTERNAL,
alert-qualifier OCTET STRING OPTIONAL }

--

auto-modify AUTO-ACTION ::= {
REGISTRATION PARAMETER IS AutoModifyRegistrationParameter
ERRORS {security-error | service-error | modify-error | message-group-error}
IDENTIFIED BY id-act-auto-modify }

AutoModifyRegistrationParameter ::= SET {
entry-class [0] EntryClass DEFAULT delivery,
filter [1] Filter OPTIONAL,
modifications [2] SEQUENCE SIZE (1..ub-modifications) OF EntryModification }

--

auto-correlate-reports AUTO-ACTION ::= {
IDENTIFIED BY id-act-auto-correlate-reports }

--

auto-delete AUTO-ACTION ::= {
ERRORS {security-error}
IDENTIFIED BY id-act-auto-delete }

-- Erreurs d'action automatique

auto-alert-error AUTO-ACTION-ERROR ::= {
PARAMETER SEQUENCE SIZE (1..ub-alert-addresses) OF AutoAlertErrorIndication
CODE global:id-aae-auto-alert-error }

AutoAlertErrorIndication ::= SET {
failing-alert-destination [0] AlertDestination OPTIONAL,
supplementary-information [1] GeneralString (SIZE (1..ub-supplementary-info-length)) OPTIONAL }

END -- fin des types d'actions automatiques générales MS

Annexe F

Récapitulation des modifications apportées aux précédentes éditions

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

La présente Définition de service diffère considérablement des versions publiées avant 1994 à la fois sur le plan rédactionnel et sur le plan technique. La présente annexe énumère les modifications techniques. Ces modifications étendent la fonctionnalité et les éléments des protocoles des versions antérieures. La présente annexe indique également les modifications minimales qu'il faut apporter à une réalisation qui prend en charge un contexte d'application 1988 pour qu'elle prenne en charge un contexte d'application 1994.

F.1 Modifications introduites dans l'édition de 1994

Les éléments de service supplémentaires pris en charge par l'édition de 1994 sont énumérés dans l'Annexe C et définis dans l'Annexe B de la Rec. UIT-T F.400/X.400 | ISO/CEI 10021-1. Les extensions aux types ASN.1 définis dans la présente Spécification sont marquées comme cela est décrit au 5.7.

Le service abstrait MS tel qu'il est spécifié dans la présente Définition de service a été élargi pour permettre d'utiliser les nouveaux services optionnels suivants avec les contextes d'application 1994:

- a) l'accès de dépôt MS a été défini. Il remplace l'accès de dépôt indirect 1988, et offre la prise en charge de l'enregistrement des messages déposés;
- b) sept nouvelles classes d'entrée ont été définies: Submission, Draft, Stored-message, Delivery-log, Submission-log, Message-log et Auto-action-log ; voir 6.3.1;
- c) la classe d'objets informationnels ATTRIBUTE et plusieurs nouveaux types d'attribut ont été définis, principalement pour être utilisés avec les entrées des nouvelles classes d'entrée; voir 6.3.3.3;
- d) quatre nouveaux types d'entrée ont été définis: submitted-message, submitted-probe, draft-message et auto-action-event; voir 6.3.6;
- e) des règles de criblage ont été définies et sont citées dans toutes les définitions d'attribut, pour indiquer les types de cribles qui peuvent être réalisés pour chaque type d'attribut; voir 6.3.9, 8.1.2.2 et l'article 12;
- f) un schéma de groupage des messages pour la classification des messages enregistrés a été introduit; voir 6.4;
- g) des erreurs d'action automatique peuvent être journalisées dans la classe d'entrées Auto-action-log; voir 6.5.3;
- h) un utilisateur peut spécifier différents ensembles d'informations de consignation pour chaque agent utilisateur utilisé (voir 8.2.5.1) et peut spécifier, dans l'argument de rattachement MS, la consignation qui doit s'appliquer à l'association abstraite en cours (voir 7.1.1);
- i) les restrictions de recherche sur la longueur du contenu s'appliquent maintenant à la longueur de tous les attributs recherchés; voir 7.1.1;
- j) un mécanisme d'extension est utilisé dans les opérations abstraites pour permettre l'ajout d'extensions propres au contenu (par exemple pour le système de messagerie de personne à personne IPMS) ou de futures extensions générales;
- k) si cela est demandé, le rattachement MS-bind signale les classes d'entrée, les règles de criblage et la profondeur des groupes de messages pris en charge. L'indication d'erreur d'action automatique et les informations de service peuvent également être signalées; voir 7.1.2;
- l) les erreurs abstraites supplémentaires qui peuvent être renvoyées par une opération abstraite ou journalisées dans la classe d'entrées Auto-action-log sont définies à l'article 9;
- m) si cela est demandé, des opérations abstraites peuvent s'appliquer aux entrées dans l'ordre inverse de leur numéro de séquence et des résultats peuvent être renvoyés à partir des entrées dans ce même ordre, voir 8.1.1, 8.1.3, 8.2.2, 8.2.3 et 8.2.7;
- n) le paramètre entry-information signale maintenant les attributs dans lesquels le nombre de valeurs dépasse la limite fixée; voir 8.1.5;
- o) l'opération abstraite Delete signale les numéros de séquence des entrées sélectionnées pour être supprimées, voir 8.2.4.2;

- p) si cela est demandé, l'opération abstraite Register-MS signale les informations consignées au moment de la demande; voir 8.2.5.2;
- q) l'opération abstraite Modify (voir 8.2.7) et l'action automatique Auto-modify (voir 13.2) peuvent être utilisées pour modifier les valeurs d'un ensemble limité d'attributs généraux et propres au contenu;
- r) l'opération abstraite de dépôt MS-message-submission prend en charge l'enregistrement des messages déposés et des brouillons de messages. L'opération abstraite de dépôt MS-probe-submission prend en charge l'enregistrement des messages d'essai; voir 8.3.1, 8.3.2, 8.1.6 et 8.1.7;
- s) l'action automatique de retransmission Auto-forward, qui a été rendue spécifique à la messagerie de personne à personne (IPM), est définie dans la Rec. UIT-T X.420 | ISO/CEI 10021-7 comme l'action automatique IPM auto-forward;
- t) l'action automatique Auto-correlate-reports corrèle les rapports remis avec les messages et les messages d'essai déposés auxquels ils sont liés, voir 13.3;
- u) l'action automatique Auto-delete supprime les entrées au bout d'une durée prédéterminée après la création de l'entrée, selon ce que l'utilisateur MS a spécifié ou ce qui a été convenu par abonnement, voir 13.4.

F.2 Modifications minimales requises pour prendre en charge les contextes d'application 1994

Les modifications minimales qu'il faut apporter à une réalisation qui prend en charge un contexte d'application 1988 pour qu'elle prenne en charge un contexte d'application 1994 sont énumérées ci-dessous. Ces prescriptions se limitent aux aspects obligatoires de la version de 1994 du service abstrait MS et supposent qu'aucune des composantes optionnelles du service de 1994 n'est prise en charge. Les aspects obligatoires suivants du service abstrait MS de 1994 doivent être pris en charge lorsqu'un contexte d'application 1994 est utilisé:

- a) la classe d'entrées Stored-message doit être prise en charge; voir 6.3.1;
- b) si un paramètre UA-registration-identifier ou un paramètre d'extensions bind-extensions est présenté dans l'argument de rattachement MS, la mémoire de messages peut renvoyer un résultat de rattachement MS qui signale que le paramètre UA-registration-identifier est inconnu ou que les extensions ne sont pas prises en charge. Si l'utilisateur MS présente le paramètre MS-configuration-request, la mémoire de messages doit signaler les classes d'entrées prises en charge; voir 7.1.1;
- c) si un intervalle est présenté dans l'ordre décroissant (voir 8.1.1), la mémoire de messages ne doit pas créer d'erreur d'intervalle (voir 9.7). Les opérations abstraites doivent être appliquées aux entrées selon l'ordre de l'intervalle et les résultats doivent être signalés dans le même ordre;
- d) la composante value-count-exceeded du paramètre entry-information doit être présente dans la cas défini au 8.1.5;
- e) si une composante selector est présente dans l'argument de l'opération abstraite Delete, le résultat de suppression doit indiquer les entrées supprimées; voir 8.2.4.2;
- f) si l'utilisateur MS présente les paramètres UA-registrations, submission-defaults ou message-group-registrations dans l'argument de consignation MS, la mémoire de messages peut renvoyer une erreur de consignation MS. Si l'utilisateur MS présente le paramètre registration-status-request, les informations demandées seront renvoyées dans le résultat de consignation; voir 8.2.5;
- g) si l'utilisateur MS invoque l'opération abstraite Modify, la mémoire de messages peut indiquer que le type d'attribut n'est pas disponible ou que la modification est inappropriée; voir 8.2.7;
- h) si l'utilisateur MS présente le paramètre submission-options dans l'argument de l'opération abstraite de dépôt MS-message-submission ou MS-probe-submission, la mémoire de messages peut renvoyer une erreur rejetant l'option demandée; voir 8.3.1.1 et 8.3.2.1;
- i) une règle de criblage d'ordonnancement doit être prise en charge pour le type d'attribut entry-type;
- j) une règle de criblage d'ordonnancement doit être prise en charge pour le type d'attribut priority, si celui-ci est pris en charge. Une règle de criblage d'égalité doit être prise en charge pour le type d'attribut proof-of-delivery-request, si celui-ci est pris en charge.

Annexe G

Définition formelle des bornes supérieures des paramètres MS

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

La présente annexe définit, à titre de référence, les bornes supérieures des divers types de données de longueur variable dont les syntaxes abstraites sont définies dans les modules ASN.1 figurant dans le corps de la présente Définition de service.

MSUpperBounds {joint-iso-itu-t mhs(6) ms(4) modules(0) upper-bounds(4) version-1994(0)}

DEFINITIONS ::=

BEGIN

-- Prologue

-- Exporte tout

IMPORTS -- rien -- ;

-- Bornes supérieures

ub-alert-addresses INTEGER ::= 16

ub-attribute-values INTEGER ::= 32767 -- (2¹⁵ – 1) l'entier le plus
-- grand représentable sur 16 bits --

ub-attributes-supported INTEGER ::= 1024

ub-auto-action-errors INTEGER ::= 32767 -- (2¹⁵ – 1) l'entier le plus
-- grand représentable sur 16 bits --

ub-auto-actions INTEGER ::= 128

ub-auto-registrations INTEGER ::= 1024

ub-default-registrations INTEGER ::= 1024

ub-entry-classes INTEGER ::= 128

ub-entry-types INTEGER ::= 16

ub-error-reasons INTEGER ::= 16

ub-extensions INTEGER ::= 32

ub-group-depth INTEGER ::= 64

ub-group-descriptor-length INTEGER ::= 256

ub-group-part-length INTEGER ::= 128

ub-information-bases INTEGER ::= 16

ub-matching-rules INTEGER ::= 1024

ub-message-groups INTEGER ::= 8192

ub-message-notes-length INTEGER ::= 1024

ub-messages INTEGER ::= 2147483647 -- (2³¹ – 1) l'entier le plus
-- grand représentable sur 32 bits --

ub-modifications INTEGER ::= 32767 -- (2¹⁵ – 1) l'entier le plus
-- grand représentable sur 16 bits --

ISO/CEI 10021-5 : 1995 (F)

ub-msstring-match	INTEGER ::= 512	
ub-per-auto-action	INTEGER ::= 32767	-- $(2^{15} - 1)$ l'entier le plus -- grand représentable sur 16 bits --
ub-per-entry	INTEGER ::= 1024	
ub-service-information-length	INTEGER ::= 2048	
ub-summaries	INTEGER ::= 16	
ub-supplementary-info-length	INTEGER ::= 256	
ub-ua-registration-identifier-length	INTEGER ::= 32	
ub-ua-registrations	INTEGER ::= 128	
ub-restrictions	INTEGER ::= 16	

END -- fin des bornes supérieures des paramètres MS

Annexe H

Groupage des messages

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

La présente annexe contient des exemples illustrant l'utilisation du groupage des messages dans la définition d'un cadre pour l'enregistrement des messages et indique des solutions possibles aux problèmes qui peuvent intervenir dans l'interfonctionnement d'agents utilisateur.

H.1 Etablissement d'un cadre d'enregistrement

Le groupage des messages permet à l'utilisateur final de définir un cadre d'enregistrement adapté aux types de message que l'utilisateur rencontre couramment. Une partition simple à un niveau de la base de données des messages en plusieurs catégories convient pour certaines applications. Pour des situations plus complexes, il peut être souhaitable de définir une structure hiérarchique de groupes de messages. Par exemple, un utilisateur pourrait d'abord partitionner sa base de données de messages en projets (par exemple projets MS, projets UA) et ensuite créer des subdivisions à l'intérieur de chaque projet (par exemple EDI-MS, IPM-MS, VM-MS). Un autre niveau de subdivision pourrait utiliser des catégories telles que planification, coûts et annonces.

L'attribut message-group-name (nom de groupe de messages) prend en charge cette affectation complexe de noms en affectant les noms des subdivisions aux composantes successives de la séquence de chaînes qui constituent le nom. Ainsi le nom de groupe de messages 'projets MS+EDI-MS+planification' se compose d'une séquence de trois parties de nom de groupe (où '+' représente un symbole de concaténation). On peut considérer que les noms de groupe de messages qui commencent par les mêmes composantes (par exemple 'projets MS+EDI-MS' et 'projets MS+IPM-MS') ont le même groupe de messages père. Cependant, cette interprétation est créée de toutes pièces par l'utilisateur MS. Il n'est pas nécessaire que la mémoire de messages elle-même définisse une telle hiérarchie à partir de cette relation implicite entre noms de groupes de messages qui partagent des composantes communes.

Un message peut appartenir à zéro, un ou plusieurs groupes de messages et peut donc apparaître à plusieurs reprises dans la hiérarchie d'enregistrement.

H.2 Introduction des messages dans le cadre d'enregistrement

Le mécanisme principal d'affectation des messages à des groupes de messages est l'action automatique Auto-modify (modification automatique). En consignnant un ensemble approprié d'actions automatiques Auto-modify, l'utilisateur final peut provoquer l'attachement d'attributs message-group-name (nom de groupe de messages) à un message lorsque ce message répond aux critères de sélection associés. Des filtres de complexité arbitraire peuvent être conçus pour répondre aux exigences d'une application donnée. On pourra créer des groupes de messages basés sur les propriétés suivantes des messages qu'ils contiendront:

- a) sujet du message de personne à personne – Cette propriété renseigne directement sur le contenu d'un message;
- b) correspondant – Le groupage par correspondant place tous les messages reçus à partir d'un utilisateur donné et tous les messages qui lui sont adressés dans un groupe de messages convenablement nommé;
- c) adressage direct – Un utilisateur peut avoir envie de faire la distinction entre les messages qui lui sont directement adressés et ceux qu'il reçoit comme résultat du développement de la liste de distribution.

L'utilisateur peut également placer manuellement les messages dans des groupes de messages à l'aide de l'opération abstraite Modify (modification). Cependant l'affectation automatique a l'avantage d'être cohérente et ne nécessite pas d'intervention de l'utilisateur autre que la consignment.

H.3 Considérations relatives à l'interfonctionnement

Un utilisateur peut se servir de plusieurs réalisations d'agent utilisateur qui diffèrent par le nombre de parties de noms de groupes de messages pris en charge. Un agent utilisateur (UA) peut alors rencontrer des noms de groupes de messages créés par un autre agent utilisateur qui contiennent un plus grand nombre de parties de noms de groupes que le nombre de parties pris en charge par l'agent utilisateur. Chaque fois que l'agent utilisateur rencontre cette possibilité (indiquée si le résultat de rattachement MS signale une profondeur de groupe de message dépassant celle prise en charge par l'agent utilisateur), il doit opérer de la façon décrite ici.

En sélectionnant les messages, l'agent utilisateur doit utiliser la règle MS-string-list-elements-match de préférence à la règle MS-string-list-match lorsqu'un criblage d'égalité est requis. De même, il doit utiliser la règle MS-single-substring-list-elements-match de préférence à la règle MS-single-substring-list-match lorsqu'un criblage de sous-chaînes est requis. Ces règles de remplacement ont pour effet de ne pas tenir compte des parties de noms de groupe supplémentaires dans la valeur enregistrée de l'attribut message-group-name (nom de groupe de message), la conséquence est que les entrées des groupes de messages subordonnés apparaissent comme des membres du groupe de messages spécifié par l'agent utilisateur. Cependant, lorsque l'agent utilisateur sélectionne des entrées dont le nom de groupe de messages contient un nombre de parties de noms de groupes égal ou inférieur au nombre de parties prises en charge par l'agent utilisateur, il doit utiliser la règle MS-string-list-match ou la règle MS-single-substring-list-match de façon à exclure les entrées qui appartiennent aux groupes subordonnés.

En traitant les résultats des opérations abstraites List (listage), Fetch (extraction), Summarize (récapitulation) et Register-MS (consignation MS), un agent utilisateur qui extrait un nom de groupe de messages contenant un plus grand nombre de parties de noms de groupes que le nombre qu'il requiert peut ne pas tenir compte des parties de noms de groupes supplémentaires. Cela peut conduire à un résultat qui contient des valeurs en double, qui doivent être écartées.

Pour supprimer une valeur de nom de groupe de messages d'un attribut message-group-name (nom de groupe de messages) lorsque l'agent utilisateur n'est pas en mesure de spécifier le nom complet, l'agent utilisateur peut spécifier la composante position de l'argument de modification qui permet d'identifier la valeur par sa position dans l'attribut. Si l'utilisateur d'un agent utilisateur qui prend en charge un nombre limité de parties de noms de groupes veut affecter des noms de groupes plus longs aux entrées, il doit utiliser un autre agent utilisateur (qui prend en charge des noms plus longs) pour consigner les actions automatiques Auto-modify (modification automatique) qui permettent d'exécuter automatiquement l'affectation.

Dans le cas contraire où la réalisation de la mémoire de messages prend en charge moins de parties de noms de groupes que l'agent utilisateur, l'agent utilisateur est obligé de ne présenter que les valeurs qui sont cohérentes avec celles qui sont prises en charge par la mémoire de messages.

La situation peut se produire lorsque l'un des agents utilisateur employés par un utilisateur se sert d'un nom de groupe de message intégré pour étiqueter les messages qui appartiennent à une certaine catégorie définie par l'application (comme 'inray', 'invisible' ou 'poubelle'). Si un autre agent utilisateur de l'utilisateur se sert d'un nom de groupe de messages intégré différent pour la même catégorie de messages, aucun des agents utilisateur ne fera l'exécution comme prévu. Ce problème peut être résolu de deux façons. Lorsque c'est possible, des attributs doivent être utilisés pour identifier les catégories standard (par exemple l'attribut marked-for-deletion (marqué pour être supprimé) doit être utilisé pour identifier la catégorie 'poubelle' de message). Autrement, lorsque aucun type d'attribut approprié n'est défini pour une certaine catégorie définie par l'application, le nom de groupe de messages utilisé pour étiqueter cette catégorie doit être rendu configurable par l'utilisateur, de sorte que tous les agents utilisateur puissent être configurés pour utiliser le même nom de groupe de message.

Annexe I

Exemple d'opération abstraite Summarize (récapitulation)

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

La présente annexe donne un exemple d'utilisation de l'opération abstraite Summarize (récapitulation).

I.1 Entrées de la mémoire de messages dans l'exemple choisi

Considérons une mémoire de messages contenant les entrées présentées dans le Tableau I.1, une entrée par ligne. Les colonnes montrent les valeurs des types d'attributs indiqués. Le caractère "-" indique que l'attribut ne figure pas dans l'entrée.

Tableau I.1 – Messages enregistrés dans l'exemple

numéro séquentiel	type d'entrée	statut d'extraction	priorité
3	message	listé	urgent
5	message	listé	faible
8	rapport	listé	-
10	message	listé	normal
15	rapport	nouveau	-
18	message	nouveau	normal
20	message	nouveau	urgent
22	message	nouveau	normal
23	message	nouveau	normal

NOTE – Même si la priorité est omise dans l'enveloppe de remise du message et que sa valeur est prise par défaut à "normal", l'attribut correspondant est présent et la valeur par défaut lui est attribuée.

I.2 Exemple de demande de récapitulation

Supposons qu'on veuille récapituler toutes les "nouvelles" entrées à l'aide de l'attribut de priorité. Le résultat demandé est la liste des décomptes figurant dans le Tableau I.2. Les nombres entre parenthèses sont les numéros de séquence des messages intervenant dans ce compte.

Tableau I.2 – Résultat attendu de l'opération abstraite Summarize (récapitulation)

priorité	dénombrement
-	1 (15)
urgent	1 (20)
normal	3 (18, 22, 23)
faible	0

Les paramètres de l'argument de récapitulation doivent être fixés de la façon suivante:

sélecteur:
 filtre: état d'extraction = nouveau
 demandes de récapitulation: type d'attribut = Priorité

Les paramètres du résultat de récapitulation pourraient être les suivants:

décompte: 5
 portée:
 borne inférieure: 15
 borne supérieure: 23
 récapitulations:
 { absent: 1
 présent: { valeur = normale, décompte = 3 }
 { valeur = urgent, décompte = 1 } }

Annexe J

Différences entre la Rec. UIT-T X.413 (1995) et l'ISO/CEI 10021-5:1995

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

La présente annexe identifie les différences connues entre la Rec. UIT-T X.413 (1994) et la Norme ISO/CEI 10021-5:1994. Il existe une différence connue:

- les parties de la notation ASN.1 spécifiant les bornes supérieures et faisant l'objet de l'annexe G ne sont pas considérées comme faisant partie intégrante l'ISO/CEI 10021-5, mais elles font partie intégrante de la Rec. UIT-T X.413.

A l'ISO/CEI, ce niveau de fonctionnalité relève de la compétence du Groupe spécial sur la normalisation fonctionnelle, qui publie des profils normalisés au niveau international (ISP). Ceux-ci peuvent spécifier les bornes supérieures des éléments protocolaires.

INDEX

Cet index donne les numéros des pages de la version anglaise dans lesquelles est défini chaque élément indexé. Il contient des éléments répartis dans les catégories suivantes:

- a) classes d'objets ASN.1
- b) modules ASN.1
- c) ensembles d'objets ASN.1
- d) objets ASN.1
- e) types ASN.1
- f) types d'attribut
- g) Définitions de paramètres
