



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.725**

(11/95)

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

---

**TECHNOLOGIES DE L'INFORMATION –  
INTERCONNEXION DES SYSTÈMES  
OUVERTS – STRUCTURE DE L'INFORMATION  
DE GESTION: MODÈLE GÉNÉRAL  
DE RELATION**

**Recommandation UIT-T X.725**

(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.725 de l'UIT-T a été approuvé le 21 novembre 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10165-7.

---

### 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, sauf ce qui est indiqué dans les notes de bas de page 4), 5) et 6) des Annexes B à D respectivement.

RECOMMANDATIONS UIT-T DE LA SÉRIE X

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

(Février 1994)

**ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X**

Domaine	Recommandations
<b>RÉSEAUX PUBLICS POUR DONNÉES</b>	
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
<b>INTERCONNEXION DES SYSTÈMES OUVERTS</b>	
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
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
<b>SYSTÈMES DE MESSAGERIE</b>	X.400-X.499
<b>ANNUAIRE</b>	X.500-X.599
<b>RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES</b>	
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
<b>GESTION OSI</b>	X.700-X.799
<b>SÉCURITÉ</b>	X.800-X.849
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
<b>TRAITEMENT OUVERT RÉPARTI</b>	X.900-X.999



## TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application.....	1
2	Références normatives .....	1
	2.1 Recommandations   Normes internationales identiques.....	1
	2.2 Recommandations   Normes internationales équivalentes par leur contenu technique.....	2
3	Définitions.....	2
	3.1 Définitions du cadre général de gestion .....	2
	3.2 Définitions de l'aperçu général de la gestion-systèmes.....	3
	3.3 Définitions du service CMIS.....	3
	3.4 Définitions du modèle d'informations de gestion.....	3
	3.5 Directives pour l'établissement de définitions d'objets gérés .....	3
	3.6 Exigences et directives concernant des formulaires de déclaration de conformité d'instance en relation avec les définitions de gestion OSI.....	4
	3.7 Définitions de la fonction de gestion d'état .....	4
	3.8 Définitions supplémentaires.....	4
4	Abréviations .....	5
5	Conventions .....	5
6	Exigences .....	5
7	Modèle .....	5
	7.1 Relations gérées .....	6
	7.2 Mappage de relation.....	8
	7.3 Spécifications réutilisables.....	9
	7.4 Représentation et gestion de relation gérée.....	9
8	Définitions génériques .....	11
	8.1 Notification et opérations de gestion des relations .....	11
	8.2 Classe d'objet géré – genericRelationshipObject .....	12
	8.3 Mappage de nom – genericRelationshipObject-system .....	12
	8.4 Attributs .....	12
	8.5 Groupe d'attributs de relation.....	12
	8.6 Paramètres.....	12
	Annexe A – Squelettes de relation .....	14
	A.1 Squelette de classe de relation .....	14
	A.2 Squelette de mappage de relation.....	18
	Annexe B – Définition de l'information de gestion.....	23
	B.1 Allocation des identificateurs d'objets .....	23
	B.2 Définition des classes d'objets gérés .....	23
	B.3 Définition des corrélations de noms.....	23
	B.4 Définition des attributs.....	23
	B.5 Définition des paramètres .....	24
	B.6 Définitions de syntaxe abstraite .....	24
	Annexe C – Formulaire de déclaration de conformité de relation gérée pour le modèle général de relation.....	25
	C.1 Introduction.....	25
	C.2 Instructions for completing the MRCS proforma .....	25
	C.3 Symbols, abbreviations and terms.....	25
	C.4 Managed relationship support.....	25

	<i>Page</i>
Annexe D – Formulaire MIDS (attributs) .....	27
D.1 Introduction.....	27
D.2 Attributes .....	27
D.3 Parameters.....	27
Annexe E – Illustration des méthodes de représentation.....	29
Annexe F – Exemples de l'utilisation de modèles .....	31
F.1 Allocation of object identifiers .....	31
F.2 Symmetric relationship example.....	31
F.3 Dependency relationship example .....	32
F.4 General composition relationship example.....	38
F.5 Access control domain example.....	39
Annexe G – Commentaires.....	43
G.1 Introduction.....	43
G.2 Dépendance entre objets gérés dans une relation gérée .....	43
G.3 Cohérence des vues.....	43
G.4 Expression d'opérations et de notifications de gestion de relation.....	43
G.5 Gestion générique .....	44
G.6 Connaissance des relations.....	44
G.7 Spécification de rôle .....	44
G.8 Réutilisation de spécifications .....	44
G.9 AND SUBCLASSES .....	44
G.10 Relations entre relations.....	44
G.11 Espace de noms des objets de relation .....	45
G.12 Méthodes de représentation permises .....	45

## **Résumé**

La présente Recommandation | Norme internationale fournit un modèle pour la définition, la représentation et la gestion de relations entre ressources, ainsi que les «outils» de notation pour spécifier ces relations. Elle spécifie également les définitions de l'information générale de gestion qui peut servir à représenter ces relations. Elle fournit enfin des directives pour l'élaboration de formulaires de déclaration de conformité. Les moyens fournis par la présente Recommandation | Norme internationale intéressent ceux qui sont impliqués dans la spécification d'un modèle d'information de gestion.



## NORME INTERNATIONALE

## RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES  
SYSTÈMES OUVERTS – STRUCTURE DE L'INFORMATION DE GESTION:  
MODÈLE GÉNÉRAL DE RELATION**

## 1 Domaine d'application

La présente Recommandation | Norme internationale fournit:

- a) un modèle de raisonnement, de représentation, de gestion et d'élaboration de spécifications réutilisables portant sur des relations entre ressources;
- b) des outils de notation servant à spécifier des relations, leur représentation et la manière dont elles sont gérées;
- c) des définitions d'une information générique de gestion pouvant être utilisées pour la représentation et la gestion de relations;
- d) des directives pour l'élaboration de formulaires de déclaration de conformité;
- e) des définitions d'exemples.

Le modèle général de relation est spécifié dans l'article 7. Les outils de notation sont spécifiés dans l'Annexe A. L'information générique de gestion est définie dans l'article 8 et dans l'Annexe B. Les directives pour la spécification de formulaires de déclaration de conformité d'instance sont données dans les Annexes C et D. Les Annexes E et F présentent respectivement des illustrations de la méthode de représentation et des définitions d'exemples. L'Annexe G contient un commentaire concernant le texte.

La présente Recommandation | Norme internationale ne fournit pas de mécanisme pour le maintien de la cohérence entre des ressources impliquées dans une relation.

La Rec. X.732 du CCITT | ISO/CEI 10164-3 spécifie un modèle de relations représentées par des attributs et un ensemble d'attributs génériques pour la représentation de types particuliers de relations. Les concepts de modélisation et les outils de spécification spécifiés dans la présente Recommandation | Norme internationale s'appliquent d'une manière générale à la définition de relations et, en conséquence, s'appliquent également à des relations représentées par des attributs telles qu'elles sont modélisées dans la Rec. X.732 du CCITT | ISO/CEI 10164-3.

## 2 Références normatives

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

### 2.1 Recommandations | Normes internationales identiques

- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040:1992, *Technologies de l'information – Interconnexion des systèmes ouverts: aperçu général de la gestion-systèmes.*
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: modèle d'information de gestion.*

- Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992) | ISO/CEI 10165-4:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: directives pour la définition des objets gérés.*
- Recommandation UIT-T X.724 (1993) | ISO/CEI 10165-6:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'instances associées à la gestion OSI.*
- Recommandation X.731 du CCITT (1992) | ISO/CEI 10164-2:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion d'états.*
- Recommandation X.732 du CCITT (1992) | ISO/CEI 10164-3:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: attributs relationnels.*

## **2.2 Recommandations | Normes internationales équivalentes par leur contenu technique**

- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*  
ISO/CEI 8824:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro 1 (ASN.1).*
- Recommandation X.291 du CCITT (1992), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications du CCITT – Spécification des suites de tests abstraites.*  
ISO/CEI 9646-2:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 2: spécification des suites de tests abstraites.*
- Recommandation UIT-T X.296 (1995), *Cadre général et méthodologie de tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Déclarations de conformité d'instance.*  
ISO/CEI 9646-7:1995, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Essais de conformité – Méthodologie générale et procédures – Partie 7: déclarations de conformité des mises en œuvre.*
- Recommandation X.700 du CCITT (1992), *Cadre de gestion pour l'interconnexion des systèmes ouverts pour les applications du CCITT.*  
ISO/CEI 7498-4:1989, *Traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: cadre général de gestion.*
- Recommandation X.710 du CCITT (1991), *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT.*  
ISO/CEI 9595:1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service commun d'informations de gestion.*
- Recommandation X.711 du CCITT (1991), *Spécification du protocole commun de transfert d'informations de gestion pour les applications du CCITT.*  
ISO/CEI 9596-1:1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Protocole commun d'information de gestion – Partie 1: spécification.*

## **3 Définitions**

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

### **3.1 Définitions du cadre général de gestion**

La présente Recommandation | Norme internationale utilise le terme ci-après, défini dans la Rec. X.700 du CCITT | ISO/CEI 7498-4:

- objet géré.

### 3.2 Définitions de l'aperçu général de la gestion-systèmes

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

- a) classe d'objets gérés;
- b) gestionnaire;
- c) déclaration de conformité d'objet géré (MOCS);
- d) formulaire de déclaration MOCS;
- e) notification;
- f) opération (de gestion-systèmes).

### 3.3 Définitions du service CMIS

La présente Recommandation | Norme internationale utilise le terme ci-après, défini dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- attribut.

### 3.4 Définitions du modèle d'informations de gestion

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

- a) action;
- b) groupe d'attributs;
- c) type d'attribut;
- d) comportement;
- e) caractéristique;
- f) confinement;
- g) héritage;
- h) invariant;
- i) héritage multiple;
- j) corrélation de nom;
- k) arbre de nomenclature;
- l) blocs de propriété (paquetages);
- m) paramètre;
- n) postcondition;
- o) précondition;
- p) spécialisation;
- q) sous-classe;
- r) objet subordonné;
- s) hyperclasse;
- t) objet supérieur.

### 3.5 Directives pour l'établissement de définitions d'objets gérés

La présente Recommandation | Norme internationale utilise les termes ci-après, définis dans la Rec. X.722 du CCITT | ISO/CEI 10165-4:

- a) définition de classe d'objet géré;
- b) squelette.

### 3.6 Exigences et directives concernant des formulaires de déclaration de conformité d'instance en relation avec les définitions de gestion OSI

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. UIT-T X.724 | ISO/CEI 10165-6:

- a) récapitulatif de conformité de relation gérée (MRCS);
- b) formulaire MRCS;
- c) déclaration de définition d'information de gestion (MIDS);
- d) formulaire MIDS.

### 3.7 Définitions de la fonction de gestion d'état

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. X.731 du CCITT | ISO/CEI 10164-2:

- a) état administratif;
- b) état opérationnel;
- c) état d'utilisation.

### 3.8 Définitions supplémentaires

**3.8.1 corrélation:** l'association d'objets gérés avec un rôle donné d'une relation gérée.

**3.8.2 prise en charge de corrélation:** capacité d'une relation gérée de prendre en charge la corrélation d'objets jouant un rôle donné pendant l'existence de la relation gérée.

**3.8.3 cohérence (d'une sous-classe):** spécialisation d'une classe de relations gérées telle que les instances de cette sous-classe puissent venir se substituer aux instances de l'hyperclasse sans affecter la fonction du système gestionnaire..

**3.8.4 relation gérée:** collection d'objets gérés ainsi qu'un invariant faisant référence aux propriétés des objets gérés.

**3.8.5 classe de relation gérée:** ensemble dénommé de relations gérées partageant la même définition.

**3.8.6 participant:** objet géré prenant en charge un rôle dans une relation gérée.

**3.8.7 pointeur de participant:** attribut identifiant des participants d'une relation gérée jouant un rôle particulier.

**3.8.8 cardinalité d'une relation:** nombre d'instances d'une même classe de relation gérée dans lesquelles un objet géré prend part avec le même rôle.

**3.8.9 contraintes de cardinalité d'une relation:** ensemble de valeurs limitant la cardinalité d'une relation.

**3.8.10 notification de gestion de relation:** notification faite par une relation gérée qui est mappée avec une ou plusieurs notifications de gestion-systèmes.

**3.8.11 opération de gestion de relation:** opération s'appliquant à une relation gérée qui est mappée avec une ou plusieurs opérations de gestion-systèmes.

**3.8.12 cardinalité de rôle:** nombre d'objets gérés participant dans un rôle donné à une relation gérée.

**3.8.13 contraintes de cardinalité de rôle:** ensemble de valeurs limitant la cardinalité d'un rôle.

**3.8.14 spécification de mappage de relation:** spécification dénommée du mappage de classes de relation gérée vers les caractéristiques d'une ou de plusieurs classes d'objets gérés.

**3.8.15 spécification d'une classe de relation:** spécification nommée des caractéristiques d'une relation gérée.

**3.8.16 rôle:** propriétés communes à une espèce donnée de participants dans une relation gérée.

**3.8.17 décorrélation:** rupture de l'association d'objets gérés avec un rôle donné dans une relation gérée.

**3.8.18 prise en charge de décorrélation:** capacité d'une relation gérée de prendre en charge la décorrélation d'objets gérés, dans un rôle donné, pendant l'existence d'une relation gérée.

## 4 Abréviations

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

ASN.1	Notation de syntaxe abstraite numéro un (voir la Rec. X.208 du CCITT   ISO/CEI 8824) ( <i>abstract syntax notation one</i> )
CMIS	Service commun de gestion d'information (voir la Rec. X.710 du CCITT   ISO/CEI 9595) ( <i>common management information service</i> )
GDMO	Directive pour la définition d'objets gérés (voir la Rec. X.722 du CCITT   ISO/CEI 10165-4) ( <i>guidelines for the definition of managed objects</i> )
MIDS	Déclaration de définition d'information de gestion ( <i>management information definition statement</i> )
MIM	Modèle d'information de gestion (voir la Rec. X.720 du CCITT   ISO/CEI 10165-1) ( <i>management information model</i> )
MRCS	Déclaration de conformité de relation gérée ( <i>managed relationship conformance statement</i> )
MOCS	Déclaration de conformité d'objet géré ( <i>managed object conformance statement</i> )
OSI	Interconnexion des systèmes ouverts ( <i>open systems interconnection</i> )
SMI	Structure de l'information de gestion ( <i>structure of management information</i> )

## 5 Conventions

Une typographie caractéristique est utilisée dans la présente Recommandation | Norme internationale lorsque le texte utilise la notation ASN.1, la notation GDMO et les outils de notation définis dans l'Annexe A.

Les outils de notation de la présente Recommandation | Norme internationale sont spécifiés conformément aux conventions définies dans la Rec. X.722 du CCITT | ISO/CEI 10165-4.

## 6 Exigences

Il peut exister des relations entre des ressources, ce qui signifie que les ressources s'affectent mutuellement. Le gestionnaire doit avoir la possibilité de gérer de telles relations, ce qui nécessite en particulier:

- un modèle de relation entre ressources indépendant de la localisation des ressources et de la méthode de représentation des relations;
- des outils de notation pour la spécification des relations;
- un modèle pour la représentation et la gestion de relations dans le contexte de la gestion-systèmes de l'OSI;
- des outils de notation pour la représentation et la gestion de relations dans le contexte de la gestion-systèmes de l'OSI;
- un modèle pour l'élaboration de spécifications réutilisables.

## 7 Modèle

Les ressources sont modélisées à des fins de gestion-systèmes sous la forme d'objets, de sorte que les relations entre ressources sont modélisées comme des **relations gérées** entre objets gérés. Une relation gérée est définie comme une collection d'objets ainsi qu'un invariant se référant aux propriétés des objets gérés. Des exemples de tels invariants sont les suivants:

- a) un objet géré doit rester dans l'état opérationnel «en service» pour remplir le rôle de fournisseur dans une relation gérée fournisseur-consommateur;

- b) un objet géré doit se trouver dans l'état opérationnel «en service», l'état d'utilisation «repos» et l'état administratif «déverrouillé» pour remplir de rôle de sauvegarde dans une relation gérée de sauvegarde;
- c) il doit exister au moins un objet géré dans le rôle de subordonné pour une relation gérée supérieur-subordonné.

Les relations gérées sont des concepts supplémentaires de modélisation de la structure de l'information de gestion, mais sont représentées et gérées par des expressions existantes du modèle de gestion de l'information (voir la Rec. X.720 du CCITT | ISO/CEI 10165-1). Des relations gérées qui partagent une définition commune sont regroupées en **classes de relation gérée**. L'Annexe A définit une notation pour la spécification de classes de relation gérée.

Le modèle reconnaît d'une manière explicite qu'une même classe de relation gérée peut être représentée de plusieurs façons dans le modèle d'information de gestion. Un **mappage de relation** décrit une représentation particulière. Les mappages de relation sont décrits en détail en 7.4. L'Annexe A définit une notation permettant de spécifier des mappages de relation gérée.

## 7.1 Relations gérées

Une relation gérée modélise sous la forme de rôles, de comportement, d'opérations et de notifications de gestion de relation, d'héritage et de propriétés caractéristiques, les propriétés d'une relation qui sont indépendantes de son mode de représentation. Les rôles sont modélisés sous la forme de classes d'objet géré compatibles, de cardinalité de rôle et de prise en charge d'opérations de corrélation et de décorrélation. Les paragraphes qui suivent détaillent les concepts de modélisation.

### 7.1.1 Opérations et notifications de gestion de relation

Les opérations et notifications de gestion de relation modélisent les opérations et notifications indépendantes du mode de représentation qui sont prises en charge par la relation gérée. Elles sont exprimées sous la forme des opérations et notifications prototypes qui suivent:

<b>ESTABLISH</b>	établir une relation gérée;
<b>TERMINATE</b>	terminer une relation gérée;
<b>BIND</b>	établir la corrélation d'un objet géré avec un rôle dans une relation gérée;
<b>UNBIND</b>	rompre la corrélation d'un objet géré avec un rôle dans une relation gérée;
<b>QUERY</b>	demander des informations concernant une relation gérée;
<b>NOTIFY</b>	rendre compte d'événements concernant une relation gérée;
<b>USER DEFINED</b>	opération définie par l'utilisateur dont la sémantique est modélisée dans le comportement de la relation gérée correspondante.

L'article 8 donne la sémantique de ces prototypes d'opérations et de notifications.

Une relation gérée modélise les opérations et notifications réelles prises en charge et toute sémantique supplémentaire par rapport à celle définie dans l'article 8.

Une relation gérée donnée peut modéliser une multiplicité d'opérations et de notifications réelles au moyen d'un prototype unique. Une relation gérée donnée peut, par exemple, modéliser une opération de décorrélation d'un participant exigeant la décorrélation et la suppression de tous les autres participants. Elle pourrait également modéliser une autre relation exigeant qu'aucun des autres participants ne soit affecté. Une relation gérée ne doit pas nécessairement modéliser une opération ou une notification de relation au sujet de chacun des prototypes.

### 7.1.2 Comportement de relation gérée

Le comportement d'une relation gérée modélise le comportement indépendant du mode de représentation d'une relation en ce qui concerne les invariants des rôles de participant et les invariants, préconditions et postconditions, concernant les opérations et notifications de gestion de relation.

**7.1.2.1 invariant:** prédicat logique qui doit rester vrai au cours d'un certain intervalle qui peut être la durée de vie de la relation gérée ou la durée d'exécution d'une opération de gestion de relation.

**7.1.2.2 précondition** (concernant une opération ou une notification de gestion de relation): prédicat logique qui doit être vrai immédiatement avant l'exécution d'une opération de gestion de relation ou immédiatement avant l'émission d'une notification de gestion de relation.

**7.1.2.3 postcondition** (concernant une opération ou une notification de gestion de relation): prédicat logique qui doit être vrai immédiatement après l'exécution d'une opération de gestion de relation ou immédiatement après l'émission d'une notification de gestion de relation.

### 7.1.3 Qualification de relation

Une qualification de relation modélise des attributs associés d'une manière intrinsèque avec la relation gérée considérée comme un tout et qui sont disponibles dans une mise en œuvre quel que soit le mode de représentation utilisé. Un appel téléphonique pourrait, par exemple, être modélisé comme une relation gérée entre deux objets gérés jouant le rôle d'abonnés, auquel cas la durée de l'appel est une propriété intrinsèque de l'appel plutôt qu'une propriété de l'un quelconque des abonnés. Toutefois, dans une mise en œuvre particulière, la durée de l'appel pourrait être mappée avec l'un des deux objets abonnés ou avec l'objet relation, en fonction du mode de représentation utilisé.

### 7.1.4 Rôles

Tout objet géré lié dans une relation gérée est identifié comme un participant et joue un ou plusieurs rôles dans la relation. Un rôle impose des exigences au sujet d'un participant et de la relation gérée. Un objet géré participant doit posséder certaines propriétés pour pouvoir jouer un rôle, une relation gérée doit se conformer aux exigences d'un rôle.

Des objets gérés appartenant à une même classe peuvent jouer des rôles différents dans une même relation gérée. Un objet géré peut jouer plus d'un rôle dans une relation gérée. Un objet géré peut participer à plus d'une instance d'une relation gérée.

#### 7.1.4.1 Propriétés des participants

Les propriétés que doit posséder un objet géré pour pouvoir jouer un rôle donné sont modélisées sous la forme d'une classe d'objet géré compatible<sup>1)</sup>. D'une manière générale, la classe compatible ne modélisera que celles des propriétés qui sont propres au rôle. Dans une mise en œuvre particulière, l'objet géré jouant le rôle peut posséder des propriétés supplémentaires, mais il doit posséder au moins les propriétés de la classe compatible et doit en conséquence être allomorphe avec la classe compatible.

#### 7.1.4.2 Cardinalité de rôle

Un certain nombre d'objets gérés peuvent en général participer dans un rôle donné à une relation gérée. Ce nombre est appelé cardinalité de rôle. Une mise en œuvre d'une relation gérée doit satisfaire à deux types de contraintes concernant la cardinalité de rôle: la cardinalité autorisée et la cardinalité exigée. Chaque contrainte est modélisée sous la forme d'un ensemble de valeurs – un ensemble de nombres entiers non négatifs dont les valeurs sont souvent consécutives.

La cardinalité de rôle autorisée est une contrainte portant sur la cardinalité de rôle qu'une mise en œuvre est autorisée à prendre en charge, alors que la cardinalité de rôle exigée est une contrainte portant sur la cardinalité dont la prise en charge est exigée pour une mise en œuvre. L'ensemble de valeurs de la cardinalité exigée doit être un sous-ensemble de la cardinalité de rôle exigée ou doit lui être égal.

#### 7.1.4.3 Prise en charge de corrélation et de décorrélation

Une relation gérée peut prendre en charge, pour chacun des rôles, l'établissement et l'annulation de corrélations d'objets gérés pendant la durée d'existence de la relation. Il s'ensuit qu'une telle relation gérée prend en charge les opérations de gestion de relation **BIND** et **UNBIND**.

Lorsqu'une relation gérée prend en charge la corrélation, des objets gérés peuvent devenir des participants dans la relation pendant l'existence de la relation, dans les limites des contraintes de cardinalité. Une tentative de violer ces contraintes provoquera l'échec de la demande **BIND**.

Lorsqu'une relation gérée prend en charge la décorrélation, des participants peuvent être retirés de la relation, dans les limites des contraintes de cardinalité. Une tentative de violer ces contraintes provoquera l'échec de la demande **UNBIND**.

<sup>1)</sup> Le concept de compatibilité est discuté au 5.2 du modèle MIM.

#### 7.1.4.4 Cardinalité de relation

Un objet géré peut participer dans le même rôle à un certain nombre d'instances d'une même classe de relation gérée. Le nombre d'instances est appelé cardinalité du rôle. Une prise en charge d'une relation gérée doit satisfaire à l'exigence d'une contrainte unique concernant la cardinalité de la relation – la contrainte de cardinalité permise pour la relation. Cette contrainte est modélisée sous la forme d'un ensemble de valeurs – un ensemble de nombres entiers non négatifs dont les valeurs sont souvent consécutives. La contrainte de cardinalité permise pour la relation limite la cardinalité d'une relation qu'une mise en œuvre peut prendre en charge.

## 7.2 Mappage de relation

Un mappage de relation modélise la représentation d'une relation gérée en ce qui concerne les propriétés d'un ou de plusieurs objets gérés, à savoir:

- le mappage de rôles de relation et de qualifications de relation avec les classes d'objet candidates;
- le mappage d'opérations et de notifications de relation respectivement avec des opérations et des notifications de gestion-systèmes;
- des objets de relation;
- des pointeurs de participant.

Il peut exister plus d'un mappage de relation associé à une classe de relation donnée.

### 7.2.1 Pointeurs de participant

Les participants à une relation et leurs rôles respectifs peuvent être identifiés au moyen d'attributs de **pointeur de participant**. La valeur d'un attribut de pointeur de participant identifie le ou les objets gérés participants, alors que le type de l'attribut identifie le rôle joué par le ou les objet(s) géré(s). Des modifications de ces attributs, réalisées soit au moyen d'opérations portant sur les attributs, soit au moyen d'opérations portant sur les objets, peuvent être utilisées pour modifier la participation des objets gérés, dans la mesure où les contraintes particulières concernant les relations gérées et le mappage des relations sont respectées. Les définitions de pointeur de participant dérivent de la définition de l'attribut **participantPointer** de l'Annexe B.

### 7.2.2 Opérations et notifications de gestion de relation

La présente Recommandation | Norme internationale ne prescrit pas de mappage des opérations et des notifications de gestion de relation avec les opérations et notifications de gestion-systèmes. L'Annexe A fournit toutefois des squelettes pour la définition de tels mappages et l'Annexe B définit des attributs utilisables pour identifier les noms de relations gérées, l'appartenance à une classe de relation et les mappages de relation en vigueur.

Les possibilités de mappage des opérations et de notifications de gestion de relation peuvent être limitées par la méthode de représentation choisie; un mappage de relation modélise le mappage pour une représentation particulière d'une classe de relation gérée. Le paragraphe 7.4 présente d'une manière plus détaillée les contraintes qu'imposent des méthodes de représentation particulières. Une opération ou notification de gestion de relation unique peut être mappée vers une ou plusieurs opérations ou notifications de gestion-systèmes.

Le mappage d'opérations ou de notifications de gestion de relation est tel que les préconditions, les postconditions et les invariants des opérations ou notifications de gestion de relation sont respectés par les opérations et notifications de gestion-systèmes. Le mappage de relation modélise le mécanisme détaillé nécessaire pour satisfaire à cette exigence. Dans l'exemple du cas d'une relation supérieur-subordonné qui exige au moins un objet géré dans le rôle subordonné, un mappage de relation pourrait modéliser le mappage de l'opération **ESTABLISH** vers:

- des opérations explicites de création des objets gérés jouant les rôles supérieur et subordonné ainsi que des opérations sur des attributs pour ajuster les pointeurs des participants; ou
- une unique opération de création de l'objet géré jouant le rôle de supérieur en comptant sur le système géré pour procéder à la création de l'objet géré jouant le rôle de subordonné et à l'ajustement des pointeurs des participants.

### 7.2.3 Comportement

Le mappage du comportement d'une relation modélise le mappage du comportement d'une relation gérée indépendant du mode de représentation, ainsi que des opérations et notifications de gestion de relation associées, vers un comportement dépendant du mode de représentation. Ceci est fait sous la forme d'invariants, de préconditions et de postconditions portant sur des opérations et notifications de gestion-systèmes liées à des participants. Le mappage du comportement modélise également tout autre comportement lié à la méthode de représentation.

### 7.3 Spécifications réutilisables

Le modèle d'élaboration de spécifications réutilisables est constitué des classes, de l'héritage et de la spécialisation de relation gérée. La caractéristique essentielle du modèle est la spécialisation, c'est-à-dire la dérivation de classes à partir de classes de relation gérée existantes au moyen de l'héritage et de spécifications incrémentales.

La spécialisation d'une classe de relation gérée peut être réalisée en combinant des caractéristiques héritées d'une ou de plusieurs classes de relations gérées avec des caractéristiques spécifiées dans le squelette de la classe de relations gérées. La classe spécialisée est appelée sous-classe de la classe ou des classes initiales. La ou les classes initiales sont appelées hyperclasses de la classe spécialisée. Les règles de spécialisation définies dans l'Annexe A assurent qu'une sous-classe est **homogène** avec sa ou ses hyperclasses. L'homogénéité d'une sous-classe avec sa ou ses hyperclasses signifie qu'une instance d'une sous-classe de relation gérée peut être substituée à une instance d'une de ses hyperclasses sans affecter le comportement d'un système gestionnaire.

### 7.4 Représentation et gestion de relation gérée

Les relations gérées peuvent être représentées au moyen des méthodes suivantes basées sur les expressions définies dans le modèle d'information de gestion (voir la Rec. X.720 du CCITT | ISO/CEI 10165-1):

- dénomination;
- pointeurs de participant;
- objets relationnels;
- opérations de gestion-systèmes.

Il n'est pas possible de représenter toutes les catégories de relations gérées au moyen de toutes les méthodes de représentation. Par exemple, le mappage d'une relation gérée avec trois rôles pourrait représenter deux des rôles au moyen de la dénomination et représenter le troisième par des pointeurs de participants.

#### 7.4.1 Représentation et gestion par dénomination

Un mappage relationnel peut modéliser la représentation et la gestion d'une relation gérée au moyen du procédé de dénomination. Un certain nombre d'objets gérés subordonnés sont désignés à l'intérieur du domaine de dénomination d'un objet géré supérieur. Le mappage relationnel indique la corrélation de nom associée à la relation gérée.

Les opérations de gestion de relation peuvent être mappées vers des opérations de gestion-systèmes portant sur l'objet géré supérieur ou sur l'objet géré subordonné. Le Tableau 1 énumère les mappages possibles. Les mappages correspondant à une représentation particulière sont modélisés par le mappage de relations correspondant.

Les participants de la relation gérée peuvent être reconnus par l'analyse des composantes du nom distinctif des subordonnés pour une corrélation de nom donnée.

**Tableau 1 – Mappage d'opération pour une relation utilisant la dénomination**

Opération de gestion de relation prototype	Opérations de gestion-systèmes candidates pour l'objet géré supérieur	Opérations de gestion-systèmes candidates pour les objets gérés subordonnés
BIND	Create, Action	Create
UNBIND	Delete, Action	Delete
QUERY	Action	Get + analyse du nom, Action
ESTABLISH	Create, Action	Create, Action
TERMINATE	Delete, Action	Delete, Action

#### 7.4.2 Représentation et gestion au moyen de pointeurs de participant

Un mappage de relation peut modéliser la représentation et la gestion d'une relation gérée au moyen d'attributs de pointeurs de participant appartenant aux participants de la relation gérée. Les opérations de gestion de relation peuvent être mappées vers des opérations d'attributs sur les pointeurs de participant. Le Tableau 2 énumère les mappages possibles. Les mappages pour une représentation particulière sont modélisés par le mappage de relation correspondant.

Tableau 2 – Mappage d'opération pour des attributs de relation

Opération de gestion de relation servant de prototype	Opérations de gestion-systèmes candidates, portant sur des attributs, appliquées aux objets gérés participants	Opérations de gestion-systèmes candidates, portant sur des objets, appliquées aux objets gérés participants
BIND	Replace, Add	Create, Action
UNBIND	Replace, Remove	Delete, Action
QUERY	Get	Action
ESTABLISH	Replace, Add	Create, Action
TERMINATE	Replace, Remove	Delete, Action

### 7.4.3 Représentation et gestion au moyen d'un objet de relation

Un mappage de relation peut modéliser la représentation et la gestion d'une relation gérée au moyen d'un objet géré appelé **objet de relation**. L'hyperclasse de toutes les classes d'objets de relation, appelée objet générique de relation (**genericRelationshipObject**) possède les attributs suivants:

- a) nom de relation – identifiant le nom de la relation gérée;
- b) classe de relation – identifiant la classe de la relation gérée;
- c) mappage de relation – identifiant le mappage de relation en vigueur.

Les classes d'objets de relation contiennent, pour chaque rôle défini dans la classe de relation, un attribut de pointeur identifiant des participants dans une instance de relation gérée.

Le groupe d'attributs de relation (voir la Rec. X.732 du CCITT | ISO/CEI 10164-3) appartient à l'objet **genericRelationshipObject**. Tous les attributs «pointeurs de participant» peuvent être placés dans ce groupe d'attributs.

Les opérations de gestion de relation sont mappées sur des opérations de gestion-systèmes portant sur l'objet de relation faisant appel à des objets ou des attributs. Le Tableau 3 donne les mappages possibles; les mappages pour une représentation particulière sont modélisés par le mappage de relation associée.

Tableau 3 – Mappage d'opération pour un objet de relation

Opération de gestion de relation prototype	Opérations de gestion-systèmes candidates, portant sur des attributs, appliquées à l'objet de relation	Opérations de gestion-systèmes candidates, portant sur des objets, appliquées à l'objet de relation
BIND	Replace, Add	Create, Action
UNBIND	Replace, Remove	Delete, Action
QUERY	Get	Action
ESTABLISH	Replace, Add	Create, Action
TERMINATE	Replace, Remove	Delete, Action

### 7.4.4 Représentation et gestion au moyen d'opérations de gestion-systèmes

Un mappage de relation peut modéliser la représentation et la gestion d'une relation gérée au moyen d'opérations de gestion-systèmes basées sur des objets portant sur les objets gérés participants. Les opérations de relation de gestion sont mappées sur les opérations de gestion-systèmes basées sur des objets portant sur les objets participants. Le Tableau 4 énumère les différentes possibilités de mappage; les mappages d'une représentation particulière sont modélisés par le mappage de la relation associée.

**Tableau 4 – Mappages d'opération pour des relations représentées par des opérations de gestion**

Opération de gestion de relation prototype	Opérations de gestion-systèmes candidates appliquées aux objets gérés participants
BIND	Create, Action
UNBIND	Delete, Action
QUERY	Action
ESTABLISH	Create, Action
TERMINATE	Delete, Action

## 8 Définitions génériques

La présente Recommandation | Norme internationale définit la sémantique de l'information de gestion générique et des opérations et notifications de gestion de relation prototypes. La spécification formelle de la syntaxe de l'information de gestion générique est donnée dans l'Annexe B.

### 8.1 Notification et opérations de gestion des relations

#### 8.1.1 ESTABLISH

Invariant: les contraintes de rôle et de cardinalité de relation sont respectées.

Précondition: la relation gérée n'existe pas; les objets gérés à lier par l'opération ESTABLISH appartiennent à une classe les autorisant à jouer le rôle.

Postcondition: la relation gérée existe; les objets gérés spécifiés dans l'opération ESTABLISH sont liés par la relation de gestion.

#### 8.1.2 TERMINATE

Précondition: la relation gérée existe.

Postcondition: la relation gérée n'existe pas; les objets qui étaient liés par la relation ne le sont plus.

#### 8.1.3 BIND

Invariant: la relation gérée existe; les contraintes de rôle et de cardinalité de relation sont respectées.

Précondition: les classes d'objets gérés spécifiés dans les opérations BIND sont celles autorisées à remplir le rôle; la relation gérée prend en charge l'opération BIND pour le rôle.

Postcondition: les objets gérés spécifiés dans l'opération BIND existent et sont liés par la relation.

#### 8.1.4 UNBIND

Invariant: la relation gérée existe et les objets gérés spécifiés dans l'opération UNBIND existent, les cardinalités de rôle et de relation sont respectées.

Précondition: les objets gérés spécifiés dans l'opération UNBIND sont liés dans la relation gérée, la relation gérée prend en charge l'opération UNBIND pour le rôle.

Postcondition: les objets gérés spécifiés dans l'opération UNBIND ne sont pas liés dans la relation gérée.

#### 8.1.5 QUERY

Précondition: «Vrai».

Postcondition: la relation gérée est inchangée.

### 8.1.6 NOTIFY

Précondition: «Vrai».

Postcondition: la relation gérée est inchangée.

### 8.1.7 USER DEFINED

La présente Recommandation | Norme internationale ne définit pas de sémantique pour ce prototype.

## 8.2 Classe d'objet géré – **genericRelationshipObject**

Toutes les classes d'objets de relation doivent être une spécialisation de l'objet **genericRelationshipObject** (*objet générique de relation*). Ceci inclut les attributs **relationshipMapping**, **relationshipClass** et **relationshipName** (*mappage de relation, classe de relation et nom de relation*). Une classe d'objet de relation pour une classe de relation donnée doit comprendre, pour chaque rôle défini dans la classe de relation gérée, un attribut dérivé de l'attribut **participantPointer** (*pointeur de participant*).

## 8.3 Mappage de nom – **genericRelationshipObject-system**

Ce mappage de nom doit être utilisé pour nommer les objets de relation en considération de l'objet système géré utilisant l'attribut **relationshipName**.

## 8.4 Attributs

### 8.4.1 **relationshipName**

Cet attribut sera utilisé pour désigner des relations gérées et pour désigner les objets de relation.

### 8.4.2 **relationshipClass**

Cet attribut sera utilisé pour identifier la classe de la relation gérée. Cette valeur sera utilisée dans le squelette correspondant de relation gérée.

### 8.4.3 **relationshipMapping**

Cet attribut sera utilisé pour identifier les mappages de relation en vigueur. Cette valeur sera utilisée dans le squelette correspondant de mappage de relation.

### 8.4.4 **participantPointer**

L'attribut de **pointeur de participant** est un attribut non enregistré qui servira de prototype pour tous les attributs de pointeur de participant. La syntaxe de cet attribut est un ensemble de noms d'objets gérés. L'attribut peut donner lieu à des tests d'égalité, de comparaison d'ensembles et d'intersection d'ensembles. L'attribut prend en charge les erreurs de violation de cardinalité de rôle, de violation de cardinalité de relation, d'absence d'un tel objet ainsi que les erreurs spécifiques de conflit d'instance de rôle.

La valeur d'un attribut de pointeur de participant dérivé indiquera le ou les objet(s) remplissant actuellement le rôle, le type de l'attribut indiquera le rôle.

## 8.5 Groupe d'attributs de relation

Cet attribut est défini dans la Rec. X.732 du CCITT | ISO/CEI 10164-3 et sera utilisé pour regrouper tous les attributs de pointeur de participant.

## 8.6 Paramètres

### 8.6.1 **noSuchObject**

Cette erreur spécifique (*pas de tel objet*) sera utilisée pour signaler qu'une opération de gestion de relation BIND (*corrélation*) a spécifié un nom d'objet géré qui n'est pas connu de l'exécutant. La valeur de ce paramètre sera le nom spécifié dans l'opération BIND.

**8.6.2 roleCardinalityViolation**

Cette erreur spécifique (*violation de cardinalité de rôle*) sera utilisée pour signaler qu'une opération de gestion de relation BIND ou UNBIND (*décorrélation*) aurait violé une des contraintes de cardinalité de rôle d'une relation gérée. La valeur de ce paramètre sera «néant».

**8.6.3 roleInstanceConflict**

Cette erreur spécifique (*conflit d'instance de rôle*) sera utilisée pour signaler qu'une opération de gestion de relation BIND a spécifié le nom d'un objet géré appartenant à une classe qui n'est pas autorisée dans le mappage de la relation gérée. La valeur de ce paramètre sera le nom spécifié dans l'opération BIND.

**8.6.4 relationshipCardinalityViolation**

Cette erreur spécifique (*violation de cardinalité de relation*) sera utilisée pour signaler qu'une opération de gestion de relation BIND ou UNBIND aurait violé une des contraintes de cardinalité de relation d'une relation gérée. La valeur de ce paramètre sera «néant».

## Annexe A

### Squelettes de relation

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

#### A.1 Squelette de classe de relation

##### A.1.1 Aperçu général

Le squelette de classe de relation est la base de la définition formelle d'une relation gérée. Les expressions figurant dans le squelette permettent de définir les différentes caractéristiques de la classe de relation gérée, à savoir:

- a) l'héritage de relation;
- b) la qualification de relation;
- c) le comportement de la relation;
- d) la compatibilité de rôle;
- e) les contraintes de cardinalité de rôle;
- f) la prise en charge de corrélation et de décorrélation;
- g) les contraintes de cardinalité de relation.

Les étiquettes et définitions auxiliaires utilisées dans le squelette de classe de relations sont définies au moyen de définitions GDMO:

<étiquette-de-comportement>  
 <étiquette-de-classe>  
 <étiquette-d'attribut>  
 identificateur d'objet  
 référence de type

Les définitions auxiliaires suivantes, utilisées dans le squelette de classe de relations, sont définies en notation ASN.1:

- identificateur

Les valeurs d'étiquettes seront uniques dans le domaine du document d'attribution.

##### A.1.1.1 Héritage

Le squelette de classe de relation gérée fournit le moyen de spécifier une ou plusieurs hyperclasses de la relation gérée à partir desquelles est dérivée la classe de relation gérée. La sous-classe hérite des caractéristiques de la ou des hyperclasse(s). La spécialisation d'une sous-classe assure que la sous-classe d'une relation gérée est consistante avec sa ou ses hyperclasse(s).

##### A.1.1.2 Qualification d'une relation

Le squelette de classe de relation gérée fournit le moyen de spécifier des caractéristiques qualifiant l'ensemble d'une relation indépendantes d'une méthode de représentation particulière.

##### A.1.1.3 Comportement

Le squelette de classe de relation gérée requiert une définition du comportement de la relation gérée qui soit indépendante de la méthode de représentation particulière. Le comportement qui dépend de la méthode de représentation particulière sera spécifié dans le squelette de mappage de la relation.

##### A.1.1.4 Rôles

Le squelette de classe de relation gérée requiert la définition des rôles de la relation et des caractéristiques qui leur sont associées.

##### A.1.1.5 Identificateur de classe de relation

Le squelette de classe de relation gérée requiert la spécification d'un identificateur d'objet pouvant être utilisé dans le protocole de gestion pour faire référence à la classe de relation.

### A.1.2 Structure du squelette

```

<relationship-class-label>  RELATIONSHIP CLASS
  [DERIVED FROM <relationship-class-label>
    [, <relationship-class-label>]* ;]
  BEHAVIOUR <behaviour-label> [, <behaviour-label>]*;
  [SUPPORTS supported [, supported]*;]
  [QUALIFIED BY <attribute-label> [, <attribute-label>]*;]
  [role-specifier]*;
REGISTERED AS object-identifier;

supporting productions

supported->
  ESTABLISH [operation-name]
  | TERMINATE [operation-name]
  | QUERY [operation-name]
  | NOTIFY [notification-name]
  | USER DEFINED [operation-name]

role-specifier->
  ROLE role-name
    [COMPATIBLE-WITH <class-label> ]
    [PERMITTED-ROLE-CARDINALITY-CONSTRAINT type-reference]
    [REQUIRED-ROLE-CARDINALITY-CONSTRAINT type-reference]
    [BIND-SUPPORT [operation-name]]
    [UNBIND-SUPPORT [operation-name]]
    [PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT type-reference]
    [REGISTERED AS object-identifier]

role-name -> <identifiant>
operation-name -> <identifiant>
notification-name -> <identifiant>

```

### A.1.3 Définitions de prise en charge

#### A.1.3.1 DERIVED FROM <relationship-class-label> [, <relationship-class-label>]\*

Cette expression (*DÉRIVE DE <étiquette-de-classe-de-relation> [, <étiquette-de-classe-de-relation>]\**) sera utilisée pour spécifier une ou plusieurs hyperclasses desquelles la classe de relation gérée hérite ses caractéristiques, y compris celles pouvant à leur tour être héritées par d'autres classes de relation gérée. La classe de relation gérée est une spécialisation des caractéristiques héritées et de celles qui sont spécifiées dans le cadre complet du squelette. La spécialisation assure que la sous-classe est consistante avec sa ou ses hyperclasses. Cette expression n'est pas présente si la classe de relation gérée n'est pas une spécialisation d'autres classes de relation.

La spécialisation de caractéristiques qui sont héritées à partir d'autres classes de relation gérée ne sera pas répétée dans la spécification des sous-classes, sauf si l'une des techniques d'extension d'une spécialisation décrites dans la Rec. X.722 du CCITT | ISO/CEI 10165-4 est utilisée.

Pour en garantir l'homogénéité, les règles de spécification des sous-classes de relations gérées sont les suivantes:

- a) **SUPPORTS**: la spécialisation des opérations de gestion de la relation gérée sera l'union des opérations de gestion des hyperclasses et de celles spécifiées dans la sous-classe; l'héritage et la spécialisation n'introduiront pas de nouvelles notifications de gestion de relation dans une sous-classe.
- b) **QUALIFIED BY**: les valeurs des ensembles de domaines d'attributs autorisés et exigés ne doivent pas être modifiées dans une sous-classe.
- c) **BEHAVIOUR**: le comportement d'une sous-classe sera le suivant:
  - la disjonction des préconditions héritées de la ou des hyperclasses et de celles spécifiées dans la sous-classe;
  - la conjonction des postconditions héritées de la ou des hyperclasses et de celles spécifiées dans la sous-classe;
  - la conjonction des invariants hérités de la ou des hyperclasses et de ceux spécifiés dans la sous-classe; la sous-classe ne peut pas être spécifiée si les invariants sont antinomiques.

d) **ROLE:**

- des spécifications de rôles supplémentaires peuvent figurer dans la définition de la sous-classe;
- une classe d'objet géré introduite dans la sous-classe au moyen de la déclaration **COMPATIBLE WITH** sera allomorphe<sup>2)</sup> de celles référencées dans des déclarations similaires de la ou des hyperclasses;
- la valeur de **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** (contrainte héritée pour la cardinalité de rôle permise) pour un rôle hérité à partir de plusieurs hyperclasses sera l'intersection des ensembles de valeurs spécifiées pour ce rôle dans chacune des hyperclasses; toute contrainte de cardinalité de rôle permise sera un sous-ensemble de la valeur de la contrainte de cardinalité de rôle permise héritée ou lui sera identique; la spécialisation de la contrainte de cardinalité de rôle permise sera l'intersection des ensembles de valeurs héritées et de celles spécifiées dans la sous-classe;
- la valeur **REQUIRED-ROLE-CARDINALITY-CONSTRAINT** (contrainte héritée pour la cardinalité de rôle exigée) pour un rôle hérité à partir de plusieurs hyperclasses sera le résultat de la réunion des ensembles de valeurs spécifiées pour ce rôle dans chacune des hyperclasses intersectées avec les valeurs de contrainte de cardinalité de rôle permise qui sont héritées, toute contrainte de cardinalité de rôle exigée sera un sous-ensemble de la valeur de la contrainte de cardinalité de rôle exigée héritée ou lui sera égale, la spécialisation de la contrainte de cardinalité de rôle exigée sera l'intersection des ensembles de valeurs héritées et de celles spécifiées dans la sous-classe;
- **BIND-SUPPORT** peut être ajouté dans la spécification de la sous-classe;
- **UNBIND-SUPPORT** peut être ajouté dans la spécification de la sous-classe;
- la valeur **PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT** (contrainte héritée pour la cardinalité relationnelle permise) pour un rôle hérité de plusieurs hyperclasses sera l'intersection logique des ensembles de valeurs spécifiées pour ce rôle dans ces hyperclasses; toute contrainte de cardinalité relationnelle permise sera un sous-ensemble au sens large de la valeur de la contrainte de cardinalité relationnelle permise héritée; la spécialisation de la contrainte de cardinalité relationnelle permise sera l'intersection des ensembles de valeurs héritées et de celles spécifiées dans la sous-classe.

e) **REGISTERED AS:** l'enregistrement de la sous-classe remplacera tout enregistrement hérité à partir d'autres définitions.

**A.1.3.2 BEHAVIOUR** <behaviour-label>[, <behaviour-label>]

Cette expression (*COMPORTEMENT* <étiquette-de-comportement> [, <étiquette de comportement>]) sera utilisée pour spécifier le comportement de la relation gérée qui est indépendant de la représentation. Elle doit être exprimée sous la forme d'un invariant de la relation gérée et sous la forme d'invariant, de préconditions et de postconditions pour les opérations et notifications de gestion de relation. Cette expression fait référence à un squelette de comportement tel qu'il est défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4.

**A.1.3.3 SUPPORTS supported** [, supported]\*

Cette expression (*PREND EN CHARGE pris-en-charge* [, *pris-en-charge*]\*) sera utilisée pour définir les opérations et notifications de gestion de relation prises en charge par une relation gérée. La production de prise en charge **supported** sera utilisée pour spécifier le prototype d'opération ou de notification servant de base à l'opération ou à la notification de gestion de relation, à savoir:

- **ESTABLISH** [nom-d-opération];
- **TERMINATE** [nom-d-opération];
- **QUERY** [nom-d-opération];
- **NOTIFY** [nom-de-notification];
- **USER DEFINED** [nom-d-opération].

<sup>2)</sup> Le concept de compatibilité est discuté au 5.2 du modèle MIM.

Le nom de l'opération et le nom de la notification doivent être utilisés, si nécessaire, afin de:

- fournir, dans le squelette de comportement référencé par la production **BEHAVIOUR**, un lien pour la spécification de tout comportement existant en plus de ceux spécifiés pour l'opération prototype référencée;
- lever l'ambiguïté pour des opérations ou notifications de gestion de relation qui sont basées respectivement sur la même opération ou notification prototype;
- fournir un lien vers les opérations et notifications de gestion de relation associées spécifiées dans le squelette de mappage de relation.

#### A.1.3.4 QUALIFIED BY <attribute-label>[, <attribute-label>]\*

Cette expression (*QUALIFIÉ PAR <étiquette-d-attribut> [, <étiquette-d-attribut>]\**) sera utilisée pour spécifier des attributs qui sont associés à l'ensemble de la relation gérée. Les attributs de qualification seront fournis par toutes les mises en œuvre de la relation gérée quelle que soit la méthode de représentation utilisée. Le squelette de mappage de relation sera utilisé pour spécifier lesquels de ces attributs seront fournis pour une représentation donnée.

#### A.1.3.5 ROLE role-name

Cette expression (*RÔLE nom-de-rôle*) sera utilisée pour spécifier les rôles associés à la classe de relation gérée, l'étiquette de nom de rôle sera utilisée comme nom de référence pour le rôle.

##### A.1.3.5.1 COMPATIBLE WITH <class-label>

Cette expression (*COMPATIBLE AVEC <étiquette-de-classe>*) sera utilisée pour spécifier les caractéristiques que doit posséder un objet géré pour remplir des exigences du rôle. Les caractéristiques seront spécifiées sous la forme d'une classe d'objet géré compatible<sup>3)</sup>. Si cette expression n'est pas présente, les caractéristiques de la classe **top** (voir la Rec. X.721 du CCITT | ISO/CEI 10165-2) sont utilisées. La spécification de rôle est indépendante de la méthode de représentation.

##### A.1.3.5.2 PERMITTED-ROLE-CARDINALITY-CONSTRAINT type-reference

Cette expression (*CONTRAİNTE-DE-CARDINALITÉ-DE-RÔLE-AUTORISÉE référence-de-type*) sera utilisée pour spécifier toute limitation du nombre d'objets gérés qu'une relation gérée peut prendre en charge dans le rôle. La contrainte sera spécifiée sous la forme d'un sous-type ASN.1 d'ensemble de valeurs d'entiers non négatifs.

Si, par exemple, l'expression spécifie un ensemble de valeurs **INTEGER (1..3)**, une relation gérée peut prendre en charge 1, 2 ou 3 objets gérés dans le rôle, mais ne peut pas en prendre en charge plus de 3. Une mise en œuvre a l'exigence de faire respecter cette contrainte.

Le rôle est optionnel si l'ensemble de valeurs contient 0. Un rôle optionnel n'implique toutefois pas la prise en charge des opérations de corrélation ou de décorrélation. Si cette expression est absente, la contrainte de cardinalité exigée du rôle obtenue par héritage est prise par défaut; dans le cas où aucune contrainte n'est héritée, l'ensemble de valeurs **INTEGER (0..MAX)** est pris par défaut.

L'ensemble de valeurs **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** doit être un surensemble de l'ensemble de valeurs **REQUIRED-ROLE-CARDINALITY-CONSTRAINT** ou être égal à cet ensemble.

##### A.1.3.5.3 REQUIRED-ROLE-CARDINALITY-CONSTRAINT type-reference

Cette expression (*CONTRAİNTE-DE-CARDINALITÉ-DE-RÔLE-EXIGÉE référence-de-type*) sera utilisée pour spécifier toute limitation du nombre d'objets gérés qu'une relation gérée doit prendre en charge dans le rôle. La contrainte sera spécifiée sous la forme d'un sous-type ASN.1 d'ensemble de valeurs d'entiers non négatifs. Si, par exemple, l'expression spécifie un ensemble d'entiers **INTEGER (1, 3, 4)**, une relation gérée doit prendre en charge 1, 3 ou 4 objets gérés dans le rôle, mais n'est pas obligée d'en prendre en charge 2 ou plus de 4. Une réalisation a l'exigence de faire respecter cette contrainte.

Le rôle est optionnel si l'ensemble de valeurs contient 0. Un rôle optionnel n'implique toutefois pas la prise en charge des opérations de corrélation ou de décorrélation. Si cette expression est absente, la contrainte de cardinalité de rôle exigée obtenue par héritage est prise par défaut; dans le cas où aucune contrainte n'est héritée, la relation gérée n'est pas obligée de respecter la continuité.

L'ensemble de valeurs **REQUIRED-ROLE-CARDINALITY-CONSTRAINT** sera toujours un sous-ensemble au sens large de l'ensemble de valeurs **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** ou égal à cet ensemble.

<sup>3)</sup> Le concept de compatibilité est discuté au 5.2 du modèle MIM.

#### A.1.3.5.4 BIND-SUPPORT [operation-name]

Cette expression (*PRISE EN CHARGE DE CORRÉLATION [nom-d-opération]*) sera utilisée pour spécifier que des objets gérés peuvent devenir des participants dans le rôle pendant l'existence de la relation dans la mesure où la cardinalité de rôle reste respectée. L'absence de cette expression implique que les participants ne peuvent devenir des participants pendant l'existence de la relation.

Le nom de l'opération doit être utilisé, si nécessaire, afin de:

- fournir, dans le squelette de comportement, un lien pour la spécification de tout comportement en plus de ceux spécifiés pour l'opération prototype **BIND**;
- lever l'ambiguïté pour des opérations multiples de gestion de relation qui sont basées sur l'opération prototype **BIND**;
- fournir un lien vers les opérations de gestion de relation associées spécifiées dans le squelette de mappage de relation.

#### A.1.3.5.5 UNBIND-SUPPORT [operation-name]

Cette expression (*PRISE EN CHARGE DE DÉCORRÉLATION [nom-d-opération]*) sera utilisée pour spécifier que des participants peuvent être libérés de leur rôle pendant l'existence d'une relation dans la mesure où la cardinalité de rôle reste respectée. L'absence de cette expression implique que les participants ne peuvent être libérés de leur rôle pendant l'existence de la relation.

Le nom de l'opération doit être utilisé, si nécessaire, afin de:

- fournir, dans le squelette de comportement, un lien pour la spécification de tout comportement en plus de ceux spécifiés pour l'opération prototype **UNBIND**;
- lever l'ambiguïté pour des opérations multiples de gestion de relation qui sont basées sur l'opération prototype **UNBIND**;
- fournir un lien vers les opérations de gestion de relation associées spécifiées dans le squelette de mappage de relation.

#### A.1.3.5.6 PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT type-reference

Cette expression (*CONTRAİNTE-DE-CARDINALITÉ-DE-RELATION-AUTORISÉE référence-de-type*) sera utilisée pour spécifier une limitation du nombre de relations de la classe donnée, dans laquelle un objet géré est autorisé à participer en jouant le rôle donné. La contrainte sera spécifiée sous la forme d'un sous-type ASN.1 d'ensemble de valeurs d'entiers non négatifs. Si par exemple l'expression spécifie un ensemble de valeurs **INTEGER (0..3)**, un objet géré peut participer dans au plus trois instances de la classe de relation gérée référencée en jouant le rôle donné. Une mise en œuvre a l'exigence de faire respecter cette contrainte. Si cette expression est absente, la contrainte de cardinalité autorisée de la relation, obtenue par héritage est prise par défaut, dans le cas où aucune contrainte n'est héritée, l'ensemble de valeurs **INTEGER (0..MAX)** est pris par défaut.

#### A.1.3.5.7 REGISTERED AS object-identifiant

Cette expression (*ENREGISTRÉ COMME identifiant d'objet*) sera utilisée pour spécifier un identifiant globalement unique qui enregistre le rôle. L'identifiant peut être utilisé pour identifier sans ambiguïté le rôle dans un protocole. Cette expression ne devra pas être présente si le rôle a été obtenu par héritage.

#### A.1.3.6 REGISTERED AS object-identifiant

Cette expression (*ENREGISTRÉ COMME identifiant d'objet*) sera utilisée pour spécifier un identifiant globalement unique qui enregistre la classe de relation gérée. L'identifiant peut être utilisé pour identifier sans ambiguïté la classe de relation gérée.

## A.2 Squelette de mappage de relation

### A.2.1 Aperçu général

Le squelette de mappage de relation est le fondement de la définition formelle d'un mappage de relation. Les expressions figurant dans le squelette permettent de définir les divers éléments de la représentation constitués par:

- a) le mappage du comportement de la relation;
- b) les objets de la relation;

- c) les classes candidates à partir desquelles les objets peuvent être créés afin de remplir des rôles;
- d) les méthodes de représentation;
- e) les attributs de qualification;
- f) les mappages de relations et de notifications.

Les étiquettes de squelette et les définitions de prise en charge suivantes, utilisées dans le squelette de mappage de relation, sont définies au moyen d'objets GDMO;

<étiquette-d-action>	<étiquette-de-corrélation-de-nom>
<étiquette-d-attribut>	<étiquette-de-notification>
<étiquette-de-comportement>	<étiquette-de-paramètre>
<étiquette-de-classe>	identificateur d'objet

La définition de prise en charge suivante, utilisée dans le squelette de mappage de relation, est définie en notation ASN.1:

- identificateur

Les valeurs d'étiquettes seront uniques au sein du document d'assignation.

### A.2.1.1 Comportement

Le squelette de mappage de relation spécifie tout comportement propre à la méthode de représentation définie dans le squelette.

### A.2.1.2 Méthodes de représentation

Le squelette de mappage de relation exige la spécification de la méthode utilisée pour représenter une relation gérée ainsi que de toute information de gestion pertinente associée à la représentation du rôle.

### A.2.1.3 Rôles

Le squelette de mappage de relation exige la spécification du mappage des rôles et des qualifications de relation vers des classes d'objets gérés.

## A.2.2 Structure du squelette

```

<relationship-mapping-label> RELATIONSHIP MAPPING
  RELATIONSHIP CLASS <relationship-class-label> ;
  BEHAVIOUR <behaviour-label> [, <behaviour-label>]*;
  [RELATIONSHIP OBJECT <class-label> [QUALIFIES <attribute-label>
    [, <attribute-label>]*];]
  role-mapping-specification [, role-mapping-specification]*;
  [OPERATIONS MAPPING relationship-operation maps-to
    [, relationship-operation maps-to ]* ;]
REGISTERED AS object-identifier;

supporting productions

role-mapping-specification ->
  ROLE role-name RELATED-CLASSES <class-label> [<class-label>]*
  [REPRESENTED-BY representation]
  [QUALIFIES <attribute-label> [ <attribute-label>]*]

representation ->
  NAMING <name-binding-label> USING superiorOrSubordinate
  | ATTRIBUTE <attribute-label>
  | RELATIONSHIP-OBJECT-USING-POINTER <attribute-label>
  | OPERATION

superiorOrSubordinate ->
  SUPERIOR|SUBORDINATE

```

relationship-operation ->

ESTABLISH [operation-name]  
 | TERMINATE [operation-name]  
 | BIND [operation-name] [role-name]  
 | UNBIND [operation-name] [role-name]  
 | QUERY [operation-name] [role-name]  
 | NOTIFY [notification-name]  
 | USER DEFINED [operation-name]

maps-to ->

MAPS-TO-OPERATION systems-management-operation  
 OF role-or-relObject [systems-management-operation  
 OF role-or-relObject]\*

systems-management-operation ->

GET <attribute-label> [<parameter-label>]\*  
 | REPLACE <attribute-label> [<parameter-label>]\*  
 | ADD <attribute-label> [<parameter-label>]\*  
 | REMOVE <attribute-label> [<parameter-label>]\*  
 | CREATE [<class-label>] [<parameter-label>]\*  
 | DELETE [<parameter-label>]\*  
 | ACTION <action-label> [<parameter-label>]\*  
 | NOTIFICATION <notification-label> [<parameter-label>]\*

role-or-relObject -> role-name | RELATIONSHIP OBJECT

role-name -> <identifiant>

operation-name -> <identifiant>

notification-name -> <identifiant>

### A.2.3 Définitions de prise en charge

#### A.2.3.1 RELATIONSHIP CLASS <relationship-class-label>

Cette expression (*CLASSE DE RELATION* <étiquette-de-classe-de-relation>) sera utilisée pour spécifier la classe de relation gérée avec laquelle ce mappage de relation est en correspondance.

#### A.2.3.2 BEHAVIOUR <behaviour-label> [, <behaviour-label>]

Cette expression (*COMPORTEMENT* <étiquette-de-comportement> [, <étiquette-de-comportement>]) sera utilisée pour spécifier le comportement de la relation gérée et de ses opérations et notifications de gestion qui est dépendant de la représentation. Le comportement doit être spécifié sous la forme d'un invariant ainsi que de préconditions et post-conditions concernant les opérations et notifications de gestion-systèmes en liaison avec les objets gérés participants. L'expression ne doit spécifier que le comportement qui est déjà présenté par les objets gérés participants.

#### A.2.3.3 RELATIONSHIP OBJECT <class-label> [QUALIFIES <attribute-label> [, <attribute-label>]\*]

Cette expression (*OBJET DE RELATION* <étiquette-de-classe> [QUALIFIE <étiquette-d-attribut> [, <étiquette-d-attribut>]\*]) figure dans des squelettes spécifiant la représentation de relation gérée au moyen d'un objet de relation. La référence <étiquette-de-classe> doit être utilisée pour spécifier la classe d'objet géré à laquelle appartient l'objet de relation. La classe d'objet géré sera une sous-classe de la classe **genericRelationshipObject** et disposera d'attributs de pointeur de participant pour chacun des rôles spécifiés dans le squelette associé de classe de relation gérée.

L'expression QUALIFIE <étiquette-de-classe> [, <étiquette-de-classe>]\* sera utilisée pour spécifier les attributs de qualification de relation définis dans le squelette de classe de relation donné, qui doivent être réalisés par l'objet de relation.

#### A.2.3.4 ROLE role-name RELATED-CLASSES <class-label> [<class-label>]\* [REPRESENTED-BY representation] [QUALIFIES <attribute-label> [, <attribute-label>]\*]

Cette expression sera utilisée pour identifier des classes d'objets gérés candidates, indiquées par le terme <étiquette-de-classe> [<étiquette-de-classe>]\*, qui peuvent remplir le rôle référencé par le **nom-de-rôle**. Le rôle sera un des rôles spécifiés dans le squelette de la classe de relation gérée concernée; les classes seront compatibles avec celles référencées par la déclaration **COMPATIBLE WITH** du même squelette. Le rôle ne pourra être assumé, dans une instance de la classe de relation gérée concernée, que par des objets appartenant aux classes spécifiées dans l'expression <étiquette-de-classe> [, <étiquette-de-classe>]\* ou à leurs sous-classes.

La définition prenant en charge la production **représentation** spécifiera la méthode utilisée pour représenter le rôle concerné ainsi que toute information de gestion associée. Il sera choisi une des productions suivantes pour spécifier respectivement une représentation par dénomination, par pointeurs de participant, par objet de relation ou par opérations de gestion-systèmes:

- **NAMING** <name-binding-label> **USING superiorOrSubordinate**: le rôle désigné par le nom-de-rôle sera représenté par un objet de la classe **SUPERIOR OBJECT CLASS** ou **SUBORDINATE OBJECT CLASS** indiquée dans la corrélation de nom <étiquette-de-corrélation-de-nom>; le développement de la production **superiorOrSubordinate** générera une déclaration **SUPERIOR** ou **SUBORDINATE**, désignant respectivement la classe d'objets supérieure ou inférieure.
- **ATTRIBUTE** <attribute-label>: le type d'attribut référencé par <étiquette-d-attribut> indiquera le rôle concerné, la valeur de l'attribut spécifiera le ou les participants qui remplissent le rôle.
- **RELATIONSHIP-OBJECT-USING-POINTER** <attribute-label>: le type d'attribut référencé par <étiquette-d-attribut> indiquera le rôle concerné, la valeur de l'attribut spécifiera le ou les participants qui remplissent le rôle.
- **OPERATION**: le mappage d'opérations de gestion de relation vers des opérations de gestion-systèmes sera spécifié dans l'expression **OPERATIONS MAPPING**.

L'expression **QUALIFIE** <attribute-label> [<attribute-label>]\* identifie des attributs qualifiant la relation, définis dans le squelette de classe de relation concerné, qui sont à réaliser par la classe d'objet géré référencée.

#### A.2.3.5 OPERATIONS MAPPING relationship-operation maps-to [, relationship-operation maps-to]\*

Cette expression (*MAPPAGE D'OPÉRATIONS opération-de-relation mappe-vers [, opération-de-relation mappe-vers]\**) sera utilisée pour spécifier le mappage d'une opération de gestion de relation vers une ou plusieurs opérations de gestion-systèmes.

La définition de prise en charge d'**opération de relation** spécifie une des productions suivantes qui sera utilisée pour indiquer l'opération ou la notification de gestion de relation correspondante ainsi que le rôle auquel elle fait référence:

- **ESTABLISH** [nom-d-opération];
- **TERMINATE** [nom-d-opération];
- **BIND** [nom-d-opération] [nom-de-rôle];
- **UNBIND** [nom-d-opération] [nom-de-rôle];
- **QUERY** [nom-d-opération] [nom-de-rôle];
- **NOTIFY** [nom-de-notification];
- **USER DEFINED** [nom-d-opération].

Le **nom d'opération** ou le **nom de notification** spécifié sera un de ceux définis dans le squelette de classe de relation concerné et constituera, si nécessaire, la liaison entre la sémantique des opérations et notifications de gestion de relation avec leur représentation interne sous forme d'opérations et de notifications de gestion-systèmes. La spécification d'un nom de rôle est optionnelle si une relation gérée définit un seul rôle.

La définition de prise en charge de **mappage** spécifie la production suivante:

- **MAPS TO OPERATION** opération-de-gestion-systèmes **OF** rôle-ou-objet-relation [opération-de-gestion-systèmes **OF** rôle-ou-objet-relation]\*

La définition de prise en charge de l'**opération de gestion-systèmes** spécifie le choix d'une ou de plusieurs des productions suivantes, dont chacune indique l'opération ou la notification de gestion-systèmes correspondante ainsi que l'information de gestion-systèmes associée; on utilisera l[<étiquette-de-paramètre>] pour spécifier tout paramètre à associer à l'opération ou à la notification de gestion-systèmes:

- **GET** <étiquette-d-attribut> [<étiquette-de-paramètre>]\* – L'attribut mentionné par <étiquette-d-attribut> spécifiera la valeur de l'attribut à rechercher.
- **REPLACE** <étiquette-d-attribut> [<étiquette-de-paramètre>]\* – L'attribut mentionné par <étiquette-d-attribut> spécifiera la valeur de l'attribut à remplacer.

- **ADD** <étiquette-d-attribut> [<étiquette-de-paramètre>]\* – L'attribut mentionné par <étiquette-d-attribut> spécifiera l'attribut auquel la valeur sera ajoutée.
- **REMOVE** <étiquette-d-attribut> [<étiquette-de-paramètre>]\* – L'attribut mentionné par <étiquette-d-attribut> spécifiera l'attribut duquel la valeur sera retirée.
- **CREATE** [<étiquette-de-classe>] [<étiquette-de-paramètre>]\* – La classe mentionnée par <étiquette-de-classe> spécifiera la classe à laquelle appartiendra l'objet géré.
- **DELETE** [<étiquette-de-paramètre>]\*.
- **ACTION** <étiquette-d-action> [<étiquette-de-paramètre>]\* – L'action mentionnée par <étiquette-d-action> spécifiera l'action à exécuter.
- **NOTIFICATION** <étiquette-de-notification> [<étiquette-de-paramètre>]\* – La notification mentionnée par <étiquette-de-notification> spécifiera la notification à émettre.

La définition de prise en charge **rôle-ou-objet-relation** spécifie des objets gérés source ou cible de l'opération de gestion-systèmes mentionnée. Le choix d'une ou de plusieurs des productions suivantes est possible pour spécifier soit l'objet géré jouant le rôle mentionné par le nom de rôle, soit l'objet en relation:

- **nom-de-rôle;**
- **RELATIONSHIP-OBJECT.**

#### **A.2.3.6 REGISTERED AS object-identifiant (identificateur-d-objet)**

Cette expression sera utilisée pour spécifier un identificateur globalement unique qui enregistre le mappage de relation. L'identificateur pourra être utilisé dans un protocole afin d'identifier sans ambiguïté le mappage de relation.

## Annexe B

### Définition de l'information de gestion<sup>4)</sup>

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

#### B.1 Allocation des identificateurs d'objets

La présente Recommandation | Norme internationale affecte les identificateurs d'objets suivants:

GRMD {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) 1}

DEFINITIONS ::= BEGIN

```

grm-Object OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) managedObjectClass(3)}
grm-Package OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) package(4)}
grm-Parameter OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5)}
grm-NameBinding OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) nameBinding(6)}
grm-Attribute OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7)}
grm-RelationshipClass OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipClass(11)}
grm-RelationshipMapping OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipMapping(12)}
grm-RelationshipRole OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipRole(13)}

```

END

#### B.2 Définition des classes d'objets gérés

```

genericRelationshipObject MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;
CHARACTERIZED BY genericRelationshipObjectPackage PACKAGE
ATTRIBUTES relationshipName GET,
relationshipClass GET,
relationshipMapping GET;
ATTRIBUTE GROUPS
"CCITT Rec. X.721 | ISO/IEC 10165-2":relationships;;;
REGISTERED AS {GRMD.grm-Object 1};

```

#### B.3 Définition des corrélations de noms

```

genericRelationshipObject-system NAME BINDING
SUBORDINATE OBJECT CLASS genericRelationshipObject AND SUBCLASSES;
NAMED BY SUPERIOR OBJECT CLASS "CCITT Rec. X.721 | ISO/IEC 10165-2":system AND SUBCLASSES;
WITH ATTRIBUTE relationshipName;
REGISTERED AS {GRMD.grm-NameBinding 1};

```

#### B.4 Définition des attributs

```

relationshipName ATTRIBUTE
WITH ATTRIBUTE SYNTAX GRM-ASN1Module.SimpleNameType;
REGISTERED AS {GRMD.grm-Attribute 1};

relationshipClass ATTRIBUTE
WITH ATTRIBUTE SYNTAX GRM-ASN1Module.RelationshipClass;
MATCHES FOR EQUALITY;
REGISTERED AS {GRMD.grm-Attribute 2};

relationshipMapping ATTRIBUTE
WITH ATTRIBUTE SYNTAX GRM-ASN1Module.RelationshipMapping;
MATCHES FOR EQUALITY;
REGISTERED AS {GRMD.grm-Attribute 3};

```

<sup>4)</sup> Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le contenu de la présente annexe pour utiliser celui-ci conformément à son objet.

participantPointer ATTRIBUTE

WITH ATTRIBUTE SYNTAX GRM-ASN1Module.GroupObjects;  
MATCHES FOR EQUALITY, SET-INTERSECTION, SET-COMPARISON;  
PARAMETERS noSuchObject,  
roleInstanceConflict,  
roleCardinalityViolation,  
relationshipCardinalityViolation;;

-- On peut choisir dans une réalisation d'appliquer les restrictions de sous-typage de l'ASN.1 à la syntaxe  
-- de l'attribut participantPointer afin de mieux refléter les contraintes de cardinalité de rôle autorisée  
-- définies dans la présente spécification.

## B.5 Définition des paramètres

noSuchObject PARAMETER

CONTEXT SPECIFIC-ERROR;  
WITH SYNTAX GRM-ASN1Module.ObjectInstance;  
REGISTERED AS {GRMD.grm-Parameter 1};

roleCardinalityViolation PARAMETER

CONTEXT SPECIFIC-ERROR;  
WITH SYNTAX GRM-ASN1Module.Null;  
REGISTERED AS {GRMD.grm-Parameter 3};

roleInstanceConflict PARAMETER

CONTEXT SPECIFIC-ERROR;  
WITH SYNTAX GRM-ASN1Module.ObjectInstance;  
REGISTERED AS {GRMD.grm-Parameter 2};

relationshipCardinalityViolation PARAMETER

CONTEXT SPECIFIC-ERROR;  
WITH SYNTAX GRM-ASN1Module.Null;  
REGISTERED AS {GRMD.grm-Parameter 4};

## B.6 Définitions de syntaxe abstraite

GRM-ASN1Module {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) 2}  
DEFINITIONS ::= BEGIN

IMPORTS ObjectInstance FROM CMIP-1 {joint-iso-itu-t ms(9) cmip(1) version(1) protocol(3) }  
SimpleNameType, GroupObjects

FROM Attribute-ASN1Module {joint-iso-itu-t ms(9) smi(3) part2(2) asn1Module(2) 1}

RelationshipClass ::= OBJECT IDENTIFIER

RelationshipMapping ::= OBJECT IDENTIFIER

Null ::= NULL

END

**Annexe C**

**Formulaire<sup>5)</sup> de déclaration de conformité de relation gérée  
pour le modèle général de relation**

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

**C.1 Introduction**

The purpose of the proforma in this annex is to provide guidelines the for Managed Relationship Conformance Statement (MRCS) so that a supplier of an implementation which claims to conform to a managed relationship class can provide conformance information in a standard form. The proforma defined in this annex is an additional proforma to that specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

**C.2 Instructions for completing the MRCS proforma**

The MRCS proforma contained in this annex is comprised of information in tabular form. The supplier of the implementation shall state which items are supported in Tables C.1 to C.3 and if necessary provide additional information.

**C.3 Symbols, abbreviations and terms**

The following common notations defined in CCITT Rec. X.291 | ISO/IEC 9646-2 are used for the status columns:

- m Mandatory
- o Optional
- c Conditional
- x Prohibited
- Not applicable

The following common notations, defined in CCITT Rec. X.291 | ISO/IEC 9646-2 and ITU-T Rec. X.296 | ISO/IEC 9646-7 are used for support columns:

- Y Implemented
- N Not implemented
- No answer required
- Ig The item is ignored (i.e. processed syntactically but not semantically)

**C.4 Managed relationship support**

The supplier of the implementation shall state the managed relationship class and the relationship mappings supported using Table C.1.

**Table C.1 – Managed relationship support**

Index	Relationship class template label	Value of object identifier for relationship class	Relationship mapping template label	Value of object identifier for relationship mapping	Status	Support	Additional information
1							

**C.4.1 Roles support**

For each role identified in the relationship mapping, the supplier of the implementation shall indicate support using Table C.2.

<sup>5)</sup> Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MRCS de la présente annexe pour utiliser celui-ci conformément à son objet.

**Table C.2 – Roles support**

Index	Role label	Constraints and values	Status	Supported	Value of object identifier for actual participants managed object class	MOCS reference for actual participants managed object class	Additional information
1							
2							

**C.4.1.1 Relationship management operations, notifications, and parameters support**

The supplier of the implementation shall indicate the relationship management operations and notifications supported using Table C.3.

The supplier of the implementation shall indicate support for the parameters, if any, specified in the relationship mapping template by using the parameter support table specified in Annex D

**Table C.3 – Relationship management operations and notifications support**

Index	Relationship management operation or notification	Systems management operation or notification	Constraints and values	Status	Support	Additional information
1						
2						

**C.4.2 Relationship object support**

The supplier of the implementation shall indicate support for the relationship object class, if any, specified in the relationship mapping template by using the MOCS proforma defined in ITU-T Rec. X.724 | ISO/IEC 10165-6 and MIDS proforma defined in Annex D. The relationship object class shall be a subclass of **genericRelationshipObject**.

## Annexe D

### Formulaire MIDS (attributs)<sup>6)</sup>

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

#### D.1 Introduction

The purpose of the proforma in this annex is to provide guidelines for the Management Information Definition Statement (MIDS) so that a supplier of an implementation which claims to conform to a managed relationship class can provide conformance information in a standard form.

#### D.2 Attributes

See Table D.1.

**Table D.1 – Attribute support**

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get		Replace	
				Status	Support	Status	Support	Status	Support
1	relationshipName	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 1}		o		m		x	
2	relationshipClass	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 2}		o		m		x	
3	roleMapping	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 3}		o		m		x	
4	participantPointer	–		o		o		o	

**Table D.1 (concluded) – Attribute support**

Index	Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	
1	–		–		–		
2	–		–		–		
3	–		–		–		
4	o		o		–		

#### D.3 Parameters

See Table D.2.

<sup>6)</sup> Les utilisateurs de la présente Recommandation | Norme internationale sont autorisés à reproduire le formulaire MIDS pour utiliser celui-ci conformément à son objet. Les instructions pour la construction du formulaire MIDS (attributs) sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 10165-6.

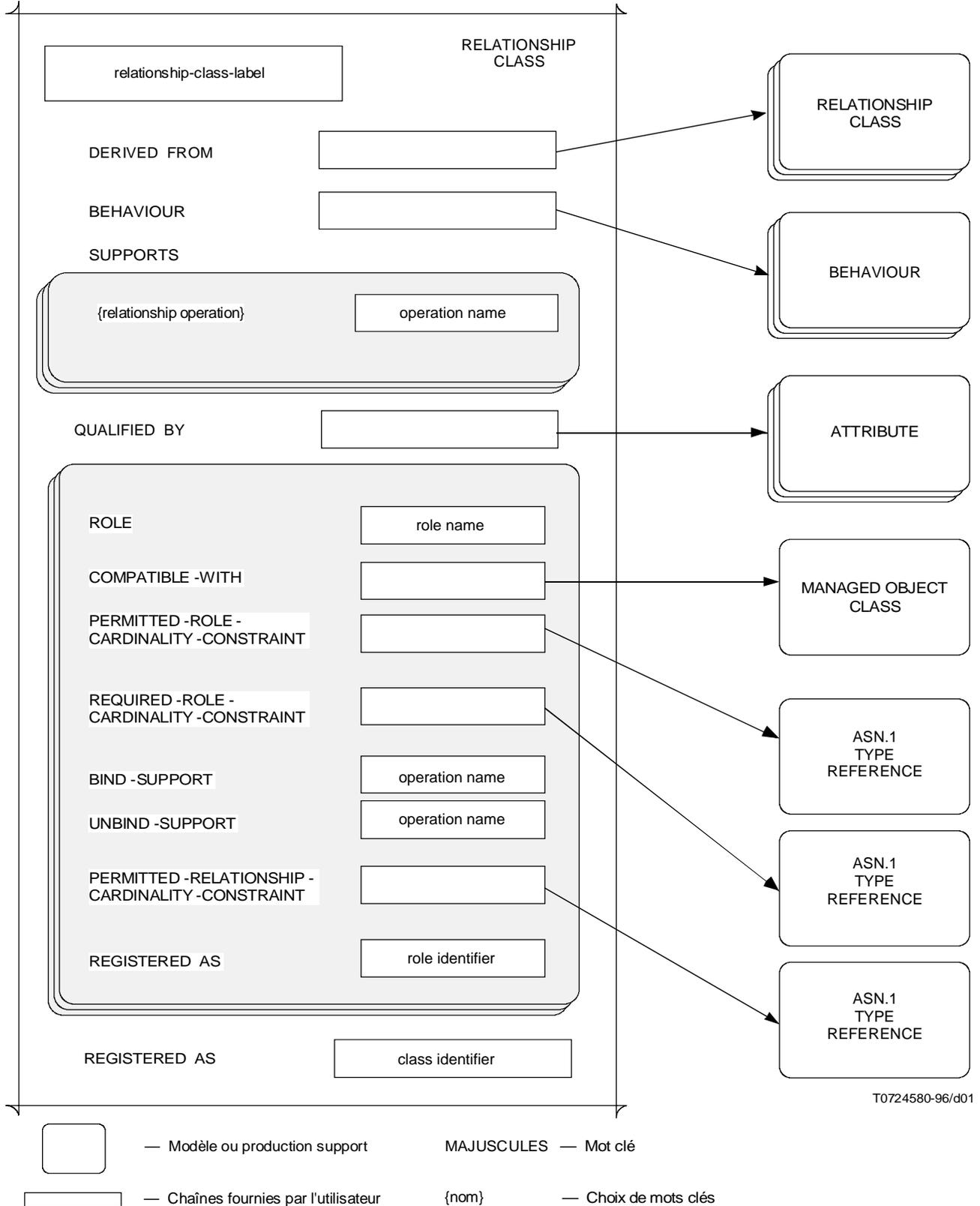
Table D.2 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	noSuchObject	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 1}		o		
2	roleCardinalityViolation	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 3}		o		
3	roleInstanceConflict	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 2}		o		
4	relationshipCardinalityViolation	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 4}		o		

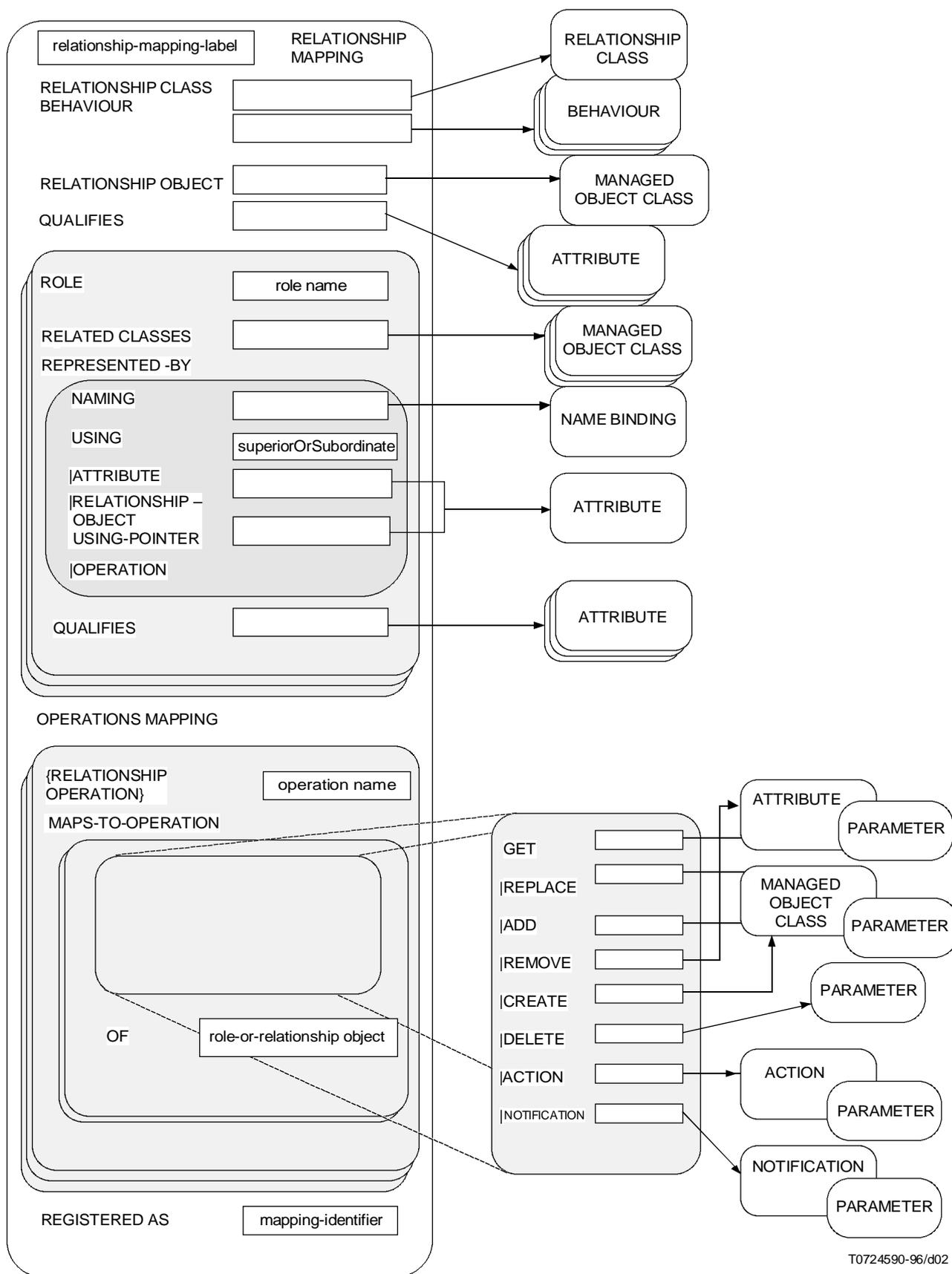
## Annexe E Illustration des méthodes de représentation

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

Cette annexe représente une interprétation graphique de la mise en page et de l'utilisation des modèles de classe de relation et de mappage de relation (voir les Figures E.1 et E.2).



**Figure E.1 – Modèle de classe de relation**



T0724590-96/d02

Figure E.2 – Modèle de mappage de relation

## Annexe F

### Exemples de l'utilisation de modèles

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

The examples shown in this annex are intended to provide illustration of the concepts identified in this Recommendation | International Standard and to give examples of the use of the **RELATIONSHIP CLASS** and **RELATIONSHIP MAPPING** template notations. These examples are not intended to provide definitions which are necessarily useful in real implementations.

#### F.1 Allocation of object identifiers

GRMExample {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) exampleASN1(99)}

DEFINITIONS ::=

BEGIN

```

    grmEx-Role OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-Role(13) exampleRole(99)}
    grmEx-RelationshipClass OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-RelationshipClass(11)exampleRelationshipClass(99)}
    grmEx-RelationshipMapping OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-RelationshipMapping(12) exampleRelationshipMapping(99)}
    grmEx-Object OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) managedObjectClass(3) exampleObjectClass(99)}
    grmEx-Attribute OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) exampleAttribute(99)}
    grmEx-NameBinding OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) nameBinding(6) exampleNameBinding(99)}
    grmEx-Package OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) package(4) examplePackage(99)}

    PersonName ::=    GraphicString
    SingleValued ::=  GroupObjects (SIZE 1)

    ZeroToTwo ::=    INTEGER (0..2)
    One ::=          INTEGER (1..1)
    OneToFive ::=   INTEGER (1..5)
    OneToMax ::=    INTEGER (1..MAX)
    Two ::=          INTEGER (2..2)
    TwoToMax ::=    INTEGER (2..MAX)

```

END

#### F.2 Symmetric relationship example

The following example shows how the relationship class template may be used to define a generic, single-role relationship between objects of the same class and how the relationship mapping template may be used to define a representation.

##### F.2.1 Symmetric relationship class definition

```

symmetricRelationship RELATIONSHIP CLASS
    BEHAVIOUR symmetricRelationshipBehaviour;

```

SUPPORTS

```

    ESTABLISH establishSymmetricRelationship,
    TERMINATE terminateSymmetricRelationship,
    QUERY      querySymmetricRelationship;

```

ROLE peerRole

```

    PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.TwoToMax
    REQUIRED-ROLE-CARDINALITY-CONSTRAINT   GRMExample.Two
    PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT GRMExample.One

```

REGISTERED AS {GRMExample.grmEx-Role x};

REGISTERED AS {GRMExample.grmEx-RelationshipClass x};

**symmetricRelationshipBehaviour BEHAVIOUR DEFINED AS "**

**INVARIANT:** This relationship has one role – the peer role – for which the minimum permitted and minimum required role cardinality constraint is 2. The existence of an instance of this relationship class implies the existence of at least two corresponding managed objects fulfilling the peer role.

**OPERATIONS:**

**ESTABLISH** establishSymmetricRelationship

**Signature:** The class and identity of the proposed participants in the peer role to be bound in a new instance of the Symmetric Relationship class.

**Precondition:** The instance of the Symmetric Relationship relationship class does not exist.

**Postcondition:** The participants in the peer role exist; the instance of the Symmetric Relationship relationship class exists; the participants in the peer role referenced in the signature are bound in this instance of the Symmetric Relationship class.

**TERMINATE** terminateSymmetricRelationship

**Signature:** The identity of the Symmetric Relationship relationship instance to be terminated.

**Precondition:** The instance of the Symmetric Relationship relationship class referenced in the signature exists; the participants in the peer role bound in this instance of the Symmetric Relationship class exist.

**Postcondition:** The referenced instance of the Symmetric Relationship relationship class does not exist; the participants in the peer role which were bound in this instance of the Symmetric Relationship class exist.";

**F.2.2 Symmetric relationship represented by a relationship object**

**symmetricRelationshipMapping RELATIONSHIP MAPPING**

**RELATIONSHIP CLASS** symmetricRelationship;

**BEHAVIOUR** symmetricRelationshipMappingBehaviour **BEHAVIOUR DEFINED AS "**

This representation of the symmetric relationship uses a relationship object to represent the relationship. Objects fulfilling the peer role are identified by the peerPointer attribute of the symmetric relationship managed object.";

**RELATIONSHIP OBJECT** symmetricRelationshipObject;

**ROLE** peerRole **RELATED-CLASSES** "CCITT Rec. X.721 | ISO/IEC 10165-2":top  
**REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER** peerPointer;

**OPERATIONS MAPPING**

**ESTABLISH** establishSymmetricRelationship

**MAPS-TO-OPERATION** CREATE OF RELATIONSHIP OBJECT,

**TERMINATE** terminateSymmetricRelationship

**MAPS-TO-OPERATION** DELETE OF RELATIONSHIP OBJECT,

**QUERY** querySymmetricRelationship

**MAPS-TO-OPERATION** GET peerPointer OF RELATIONSHIP OBJECT;

**REGISTERED AS** {GRMExample.grmEx-RelationshipMapping x};

**symmetricRelationshipObject MANAGED OBJECT CLASS**

**DERIVED FROM** genericRelationshipObject;

**CHARACTERIZED BY** symmetricRelationshipPackage **PACKAGE**

**ATTRIBUTES** peerPointer **GET-REPLACE** ADD-REMOVE;;;

**REGISTERED AS** {GRMExample.grmEx-Object x};

**peerPointer ATTRIBUTE**

**DERIVED FROM** participantPointer;

**REGISTERED AS** {GRMExample.grmEx-Attribute x};

**F.3 Dependency relationship example**

The following example illustrates a relationship of dependency of one or more objects that assume a dependency role on a single object that assumes a parent role. The example illustrates mappings in terms of participant pointers, a relationship object, and naming.

The dependency relationship class might be useful to represent a directed acyclic graph by means of relationship specialization. In such a DAGDependency relationship class, the level of a dependent relative to its parent in the graph should be introduced and represented by the addition of an appropriate attribute. An invariant should be added stating that the value of the level attribute in a dependent must always be greater than that in its parents. The dependency relationship class might also be useful to represent a family relationship by the specialization of the person managed object class into three subclasses:

- parent;
- son; and
- daughter.

### F.3.1 Dependency relationship class definition

**dependency RELATIONSHIP CLASS**

**BEHAVIOUR** dependencyBehaviour;

**SUPPORTS**

**ESTABLISH** establishDependency,  
**TERMINATE** terminateDependency,  
**QUERY** queryDependents,  
**QUERY** queryParent;

**QUALIFIED-BY** timeOfEstablishment;

**ROLE** parentRole

**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.One  
**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.One  
**PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT** GRMExample.One

**REGISTERED AS** {GRMExample.grmEx-Role x},

**ROLE** dependentRole

**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToMax  
**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.One  
**PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT** GRMExample.One

**BIND-SUPPORT** bindDependent  
**UNBIND-SUPPORT** unbindDependent

**REGISTERED AS** {GRMExample.grmEx-Role x};

**REGISTERED AS** {GRMExample.grmEx-RelationshipClass x};

**dependencyBehaviour BEHAVIOUR DEFINED AS "**

**INVARIANT:** There exist two roles in this relationship class – parent role and dependent role. The existence of a participant in the dependent role implies the existence of at least one corresponding participant in the parent role. A managed object may not participate in both roles.

**COMMENTS:** An object instance fulfilling the dependent role may only participate in one instance of this dependency relationship, that is, the relationship cardinality is equal to one. An object instance able to fulfil the parent role may exist outside a dependency relationship an object fulfilling a dependent role shall not. The qualifying attribute, timeOfEstablishment, indicates the time, in UTC time format, of establishment of the relationship.

**OPERATIONS:**

**ESTABLISH** establishDependency

**Signature:** The class and identity of the proposed participant object in the dependent role to be created by the ESTABLISH operation; the class and identity of the proposed participant in the parent role.

**Precondition:** The proposed participant in the dependent role does not exist; the proposed participant in the parent role exists.

**Postcondition:** A new instance of the dependency relationship class exists; the participants in the parent role and the dependent role proposed in the ESTABLISH signature exist and are bound in the new instance of the dependency relationship class. The qualifying attribute, timeOfEstablishment, is set to the current value of UTC time.

**BIND bindDependent**

- Signature: The class and identity of a participant in the parent role; the class and identity of the proposed participant to be created in the dependent role.
- Precondition: The participant in the parent role exists and is bound into an instance of the dependency relationship class; the proposed participant in the dependent role does not exist.
- Postcondition: The participant in the dependent role referenced in the BIND signature exists and is bound into the same dependency relationship as that in which the participant in the parent role referenced in the BIND signature is bound.

**UNBIND unbindDependent**

- Signature: The class and identity of a participant in the parent role; the class and identity of the participant in a dependent role.
- Precondition: The two participants identified in the UNBIND signature exist and are bound into the same instance of a dependency relationship; the dependency relationship exists; there exists at least one other participant in the dependent role bound into the relationship.
- Postcondition: The participant in the dependent role referenced in the UNBIND signature does not exist; all other participants bound into the instance of dependency relationship class exist and remain bound in the instance of the dependency relationship class.

**TERMINATE terminateDependency**

- Signature: The identity of a dependency relationship instance to be terminated.
- Precondition: The instance of the dependency relationship class identified in the signature exists; only a single participant in dependent role is bound into the identified dependency relationship.
- Postcondition: The instance of the dependency relationship class referenced in the signature does not exist; the participant that was in the parent role exists. The participant(s) in the dependent role do not exist. The value of the qualifying attribute, timeOfEstablishment, is undefined.";

**person MANAGED OBJECT CLASS**

**DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;**

**CHARACTERIZED BY**

**personPackage PACKAGE**

**ATTRIBUTES personName GET;;;**

**REGISTERED AS {GRMExample.grmEx-Object x};**

**personName ATTRIBUTE**

**WITH ATTRIBUTE SYNTAX GRMExample.PersonName;**

**REGISTERED AS {GRMExample.grmEx-Attribute x};**

**timeOfEstablishment ATTRIBUTE**

**WITH ATTRIBUTE SYNTAX UTCTime;**

**REGISTERED AS {GRMExample.grmEx-Attribute x};**

**F.3.2 Dependency relationship class represented by means of conjugate pointers**

**dependencyAttributeRepresentation RELATIONSHIP MAPPING**

**RELATIONSHIP CLASS dependency;**

**BEHAVIOUR dependencyAttributeRepresentationBehaviour;**

**ROLE parentRole**

**RELATED-CLASSES aPerson**

**REPRESENTED-BY ATTRIBUTE parent**

**QUALIFIES timeOfEstablishment,**

**ROLE dependentRole**

**RELATED-CLASSES bPerson**

**REPRESENTED-BY ATTRIBUTE dependents;**

**OPERATIONS MAPPING**

**ESTABLISH**    **establishDependency**  
**MAPS-TO-OPERATION CREATE OF dependentRole,**  
**TERMINATE**    **terminateDependency**  
**MAPS-TO-OPERATION DELETE OF dependentRole,**  
**BIND**            **bindDependent**  
**MAPS-TO-OPERATION CREATE OF dependentRole ,**  
**UNBIND**        **unbindDependent**  
**MAPS-TO-OPERATION DELETE OF dependentRole,**  
**QUERY**         **queryParent parentRole**  
**MAPS-TO-OPERATION GET parent OF dependentRole,**  
**QUERY**         **queryDependents dependentRole**  
**MAPS-TO-OPERATION GET dependents OF parentRole;**

**REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};**

**dependencyAttributeRepresentationBehaviour    BEHAVIOUR DEFINED AS "**

This representation of the dependency relationship class uses conjugate participant pointers to represent an instance of the relationship; participant pointer consistency is to be maintained.

The relationship management operations **ESTABLISH** **establishDependency** and **BIND** **bindDependent** both map to a create of a participant in the dependent role: the distinction being that the relationship management operation **ESTABLISH** **establishDependency** is used when a participant is the first to fulfil the dependent role; the relationship management operation **BIND** **bindDependent** is used when there is at least one other participant in the dependent role at the time of binding. After creation of an object of class **bPerson** with the attribute, **parent**, identifying an object of class **aPerson**, the value of the attribute, **dependents**, in the object of class **aPerson** identifies the corresponding object of class **bPerson**.

Similarly, the relationship management operations **TERMINATE** **terminateDependency** and **UNBIND** **unbindDependent** both map to a delete of a participant in the dependentRole: the distinction being that the relationship management operation **TERMINATE** **terminateDependency** is used if there is only one participant fulfilling the dependentRole; the relationship management operation **UNBIND** **unbindDependent** is used if there is more than one participant fulfilling the dependentRole at time of deletion. Upon deletion of an object of class **bPerson** fulfilling the dependentRole, the value of the attribute, **dependents**, in the object of class **aPerson** object fulfilling the parentRole is modified by removing the identity of the corresponding object of class **bPerson**.

The **QUERY** **queryDependents** relationship management operation maps to a **GET** of the **dependents** attribute in the **aPerson** object fulfilling a parentRole; the **QUERY** **queryParent** operation maps to a **GET** of the **parent** attribute in the **bPerson** object fulfilling a dependentRole.

The creation of a **bPerson** managed object class (or **bPerson** subclass) results in the establishment of an instance of the dependency relationship with **dependencyAttributeRepresentation RELATIONSHIP MAPPING** when the value of the **parent** attribute in the object of class **bPerson** is set-by-create to an instance of a managed object of class **aPerson** and the value of the **dependents** attribute in the object of class **aPerson** is an empty set.

The creation of a **bPerson** (or **bPerson** subclass) managed object results in its being bound to an instance of the dependency relationship with **dependencyAttributeRepresentation RELATIONSHIP MAPPING** when the value of the **parent** attribute in the **bPerson** object is set-by-create to an instance of **aPerson** managed object class and the **dependents** attribute in the **aPerson** object is a non-empty set.

The deletion of a **bPerson** (or **bPerson** subclass) managed object results in its being unbound from an instance of the dependency relationship with **dependencyAttributeRepresentation RELATIONSHIP MAPPING** when the value of the **dependents** attribute in the **aPerson** object is not empty after the deletion and associated update of the **dependents** attribute.

The deletion of a **bPerson** (or **bPerson** subclass) managed object results in the termination of an instance of the dependency relationship with **dependencyAttributeRepresentation RELATIONSHIP MAPPING** when the value of the **dependents** attribute in the **aPerson** object is empty after the deletion and the associated update of the **dependents** attribute.";

**aPerson    MANAGED OBJECT CLASS**

**DERIVED FROM** **person;**  
**CHARACTERIZED BY** **parentPackage    PACKAGE**  
**ATTRIBUTES**        **dependents GET,**  
                          **timeOfEstablishment GET;;;**

**REGISTERED AS {GRMExample.grmEx-Object x};**

**bPerson MANAGED OBJECT CLASS**

**DERIVED FROM** person;

**CHARACTERIZED BY**

dependentPackage PACKAGE

ATTRIBUTES parent PERMITTED VALUES GRMExample.SingleValued GET;;;

**REGISTERED AS** {GRMExample.grmEx-Object x};

**dependent ATTRIBUTE**

**DERIVED FROM** participantPointer;

**REGISTERED AS** {GRMExample.grmEx-Attribute x};

**parent ATTRIBUTE**

**DERIVED FROM** participantPointer;

**REGISTERED AS** {GRMExample.grmEx-Attribute x};

**F.3.3 Dependency relationship class represented by means of a relationship object**

**dependencyObjectRepresentation RELATIONSHIP MAPPING**

**RELATIONSHIP CLASS** dependency;

**BEHAVIOUR** dependencyObjectRepresentationBehaviour;

**RELATIONSHIP OBJECT** dependencyRelationshipObject

**QUALIFIES** timeOfEstablishment;

**ROLE** parentRole

**RELATED-CLASSES** person

**REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER** parent,

**ROLE** dependentRole

**RELATED-CLASSES** person

**REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER** dependents;

**OPERATIONS MAPPING**

**ESTABLISH** establishDependency

**MAPS-TO-OPERATION CREATE OF RELATIONSHIP OBJECT,**

**TERMINATE** terminateDependency

**MAPS-TO-OPERATION DELETE OF RELATIONSHIP OBJECT,**

**BIND** bindDependent dependentRole

**MAPS-TO-OPERATION ADD dependents OF RELATIONSHIP OBJECT,**

**UNBIND** unbindDependent dependentRole

**MAPS-TO-OPERATION REMOVE dependents OF RELATIONSHIP OBJECT,**

**QUERY** queryDependents dependentRole

**MAPS-TO-OPERATION GET dependents OF RELATIONSHIP OBJECT,**

**QUERY** queryParents parentRole

**MAPS-TO-OPERATION GET parent OF RELATIONSHIP OBJECT;**

**REGISTERED AS** {GRMExample.grmEx-RelationshipMapping x};

**dependencyObjectRepresentationBehaviour BEHAVIOUR DEFINED AS"**

This representation of the dependency relationship uses a relationship object to represent an instance of the relationship and to relate the participants. The relationship management operation ESTABLISH establishDependency maps to a CREATE of a dependencyRelationshipObject object and the relationship management operation TERMINATE terminateDependency maps to a DELETE of the dependencyRelationshipObject object. The relationship management operation BIND bindDependent maps to an ADD operation on the dependents attribute in a dependencyRelationshipObject object. The relationship management operation UNBIND unbindDependent maps to a REMOVE operation on the dependents attribute in the dependencyRelationshipObject object.

The creation of a DependencyRelationshipObject object results in the establishment of a dependency relationship with the dependencyObjectRepresentation RELATIONSHIP MAPPING. Because the parent role is not dynamic (i.e. BIND-SUPPORT and UNBIND-SUPPORT are not defined for the parent role), the parent attribute within the DependencyRelationshipObject must be set-by-create to the value of exactly one instance of person object fulfilling the parentRole; the value of the parent attribute cannot be changed during the lifetime of the dependency operation.

The addition of a value representing a person object to the dependents attribute of a dependencyRelationshipObject object results in the person object's being bound into the relationship corresponding to the dependencyRelationshipObject object in the dependentRole.

The removal of a value representing a person object from the dependents attribute of a dependencyRelationshipObject object, results in the person object's being unbound from the relationship corresponding to the dependencyRelationshipObject object.

The deletion of a dependencyRelationshipObject results in the termination of the corresponding dependency relationship with the dependencyObjectRepresentation RELATIONSHIP MAPPING.";

```

dependencyRelationshipObject    MANAGED OBJECT CLASS
DERIVED FROM genericRelationshipObject;
CHARACTERIZED BY
dependencyRelationshipObjectPackage PACKAGE
ATTRIBUTES
    depends GET-REPLACE ADD-REMOVE,
    parent GET,
    timeOfEstablishment GET;
REGISTERED AS {GRMExample.grmEx-Package x};;
REGISTERED AS {GRMExample.grmEx-Object x};

```

#### F.3.4 Dependency relationship represented by means of naming

```

dependencyNamingRepresentation RELATIONSHIP MAPPING
RELATIONSHIP CLASS dependency;
BEHAVIOUR dependencyNamingRepresentationBehaviour ;

ROLE parentRole
    RELATED-CLASSES cPerson
    REPRESENTED-BY NAMING aNameBinding USING SUPERIOR
    QUALIFIES timeOfEstablishment,

ROLE dependentRole
    RELATED-CLASSES person
    REPRESENTED-BY NAMING aNameBinding USING SUBORDINATE;

OPERATIONS MAPPING
ESTABLISH establishDependency
    MAPS-TO-OPERATION CREATE OF dependentRole,
BIND bindDependent dependentRole
    MAPS-TO-OPERATION CREATE OF dependentRole,
UNBIND unbindDependent dependentRole
    MAPS-TO-OPERATION DELETE OF dependentRole,
TERMINATE terminateDependency
    MAPS-TO-OPERATION DELETE OF dependentRole,
QUERY queryDependents dependentRole MAPS-TO-OPERATION GET
    "CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF dependentRole,
QUERY queryParent parentRole MAPS-TO-OPERATION GET
    "CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF dependentRole;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};

dependencyNamingRepresentationBehaviour BEHAVIOUR DEFINED AS "

```

This representation of the dependency relationship uses naming to represent an instance of the relationship.

The relationship management operations ESTABLISH establishDependency and BIND bindDependent both map to a create of a person (or person subclass) object participant in the dependentRole using a name binding with a cPerson (or cPerson subclass) object as the superior object in the parentRole. The distinction is that: the relationship management operation ESTABLISH establishDependency is used when the proposed participant in the dependent role would be the first object in the role; the relationship management operation BIND bindDependent is used when there is at least one other participant in the dependent role at the time of creation.

Similarly, the relationship management operations TERMINATE terminateDependency and UNBIND unbindDependent both map to a delete of a participant in the dependent role, the distinction being that: the relationship management operation TERMINATE terminateDependency is used if the participant is the only one fulfilling the dependentRole and the relationship management operations UNBIND unbindDependent is used if at least one other participant remains fulfilling the dependent role after deletion.

The QUERY queryDependents relationship management operation maps to a scoped get of the nameBinding attribute with a scope level of one on the person object in the parent role to determine the contained person objects that have the value of their name binding attribute equal to aNameBinding; such objects are fulfilling the dependents role.

The QUERY queryParent relationship management operation maps to a get of the nameBinding attribute of the subordinate object to determine that the value of its name binding attribute is equal to aNameBinding; subsequent analysis of the RDN of the subordinate object name will indicate the parent object.

The creation of a person (or person subclass) managed object as a subordinate to a cPerson (or cPerson subclass) object with aNameBinding name binding results in the establishment of an instance of the dependency relationship with dependencyNamingRepresentation RELATIONSHIP MAPPING if there are no other subordinates with aNameBinding name binding.

The creation of a person (or person subclass) managed object as a subordinate of a cPerson (or cPerson subclass) object with aNameBinding name binding results in the binding of the created object into a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING if there is at least one other subordinate with aNameBinding name binding.

The deletion of a person (or person subclass) managed object bound in the dependent role of a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING, results in the unbinding of the deleted object from the dependency relationship when at least one other dependents with aNameBinding will exist after the deletion.

The deletion of a person (or person subclass) managed object bound in the dependent role of a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING, results in the termination of the dependency relationship when there will exist no other dependents with aNameBinding after the deletion.";

**aNameBinding NAME BINDING**

**SUBORDINATE OBJECT CLASS person AND SUBCLASSES;**  
**NAMED BY SUPERIOR OBJECT CLASS cPerson AND SUBCLASSES;**  
**WITH ATTRIBUTE personName;**  
**CREATE;**  
**DELETE;**

**REGISTERED AS {GRMExample.grmEx-NameBinding x};**

**cPerson MANAGED OBJECT CLASS**

**DERIVED FROM person;**  
**CHARACTERIZED BY**  
**timePackage PACKAGE**  
**ATTRIBUTES timeOfEstablishment GET;;;**

**REGISTERED AS {GRMExample.grmEx-Object x};**

**F.4 General composition relationship example**

This example illustrates the use of the relationship class template to define a generic composition relationship between a single object in a composite role and one or more objects in a component role and how the template might be refined. Such a relationship might be useful for modelling an assembly/sub-assembly relationship.

**generalCompositionRelationship RELATIONSHIP CLASS**  
**BEHAVIOUR generalCompositionRelationshipBehaviour;**

**SUPPORTS**  
**ESTABLISH establishGeneralComposition,**  
**TERMINATE terminateGeneralComposition;**

**ROLE compositeRole**  
**PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne**  
**REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne**  
**REGISTERED AS {GRMExample.grmEx-Role x},**

**ROLE componentRole**  
**PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToMax**  
**REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne**  
**BIND-SUPPORT bindComponent**  
**UNBIND-SUPPORT unbindComponent**

**REGISTERED AS {GRMExample.grmEx-Role x};**  
**REGISTERED AS {GRMExample.grmEx-RelationshipClass x};**

**generalCompositionRelationshipBehaviour BEHAVIOUR DEFINED AS "**

**INVARIANT:** The existence of an instance of this relationship class implies the existence of exactly one participant in the composite role and one or more participants in the component role. At least one property of the composite participant is such that it depends upon properties of the components. At least the identity of the composite participant is such that it is independent of the existence or properties of the components; that is, creating, updating, or deleting any component does not change the identity of the composite.

## OPERATIONS:

## ESTABLISH establishGeneralComposition

- Signature: The class and identity of the proposed participant in the composite role and the class and identity of the proposed participant(s) in the component role to be bound in an instance of the generalCompositionRelationship.
- Precondition: The proposed participants are not already bound in the same instance of the generalCompositionRelationship class or a subclass thereof.
- Postcondition: An instance of the generalCompositionRelationship class exists; the participants referenced in the signature are bound into this instance of the generalCompositionRelationship class.

## BIND bindComponent

- Signature: The class and identity of a proposed participant in the component role; the identity of a generalCompositionRelationship.
- Precondition: The referenced instance of the generalCompositionRelationship class exists; the proposed participant in the component role is not bound into this instance of generalCompositionRelationship class; there exists at least one other participant in the component role bound into this instance of the generalCompositionRelationship class.
- Postcondition: The participant in the component role referenced in the signature exists and is bound in this instance of the generalCompositionRelationship class.

## UNBIND unbindComponent

- Signature: The class and identity of a participant in the component role; the identity of a generalCompositionRelationship.
- Precondition: The instance of the generalCompositionRelationship class referenced in the signature exists; the participant in the component role referenced in the signature is bound into the referenced instance of generalCompositionRelationship class; there exists at least one other participant in the component role bound into the referenced instance of the generalCompositionRelationship class.
- Postcondition: The referenced participant in the component role exists but is not bound into the referenced instance of the generalCompositionRelationship class; the referenced instance of the generalCompositionRelationship class exists.

## TERMINATE terminateGeneralComposition

- Signature: The identity of a generalCompositionRelationship instance.
- Precondition: The referenced instance of the generalCompositionRelationship class exists.
- Postcondition: The referenced instance of the generalCompositionRelationship class does not exist; the participants in the composite role and in the component role that were bound into the relationship exist.";

**F.4.1 Subclass of general composition relationship**

**subclassedCompositionRelationship RELATIONSHIP CLASS**  
**DERIVED FROM generalCompositionRelationship;**  
**BEHAVIOUR subclassedCompositionRelationshipBehaviour**  
**BEHAVIOUR DEFINED AS"**

This relationship class refines the required role cardinality of the component role of the generalCompositionRelationship class to be the range 1 to 5; all other characteristics of this relationship class are inherited from the generalCompositionRelationship class.";

**ROLE componentRole**

**REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToFive;**  
**REGISTERED AS {GRMExample.grmEx-Object x};**

**F.5 Access control domain example**

**accessControlDomain RELATIONSHIP CLASS**  
**BEHAVIOUR accessControlDomainBehaviour BEHAVIOUR DEFINED AS"**

This relationship class binds managed objects which are subject to access control (memberObjectRole) to managed objects representing the access enforcement function (aefRole) and access decision function (adfRole) respectively.";;

**SUPPORTS QUERY** queryAccessControlDomain;

**ROLE** memberObjectRole

**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToTwo

**BIND-SUPPORT** bindMember

**UNBIND-SUPPORT** unbindMember

**REGISTERED AS** {GRMExample.grmEx-Role memberObjectRoleArc(1) },

**ROLE** aefRole

**COMPATIBLE-WITH** "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter

**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne

**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne

**REGISTERED AS** {GRMExample.grmEx-Role aefRoleArc(2) },

**ROLE** adfRole

**COMPATIBLE-WITH** "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules

**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne

**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne

**REGISTERED AS** {GRMExample.grmEx-Role adfRoleArc(3) };

**REGISTERED AS** {GRMExample.grmEx-RelationshipClass accessControlDomainArc(1) };

### F.5.1 Access control domain relationship represented by attributes and naming

**simpleAccessControlDomain RELATIONSHIP MAPPING**

**RELATIONSHIP CLASS** accessControlDomain;

**BEHAVIOUR** simpleAccessControlDomainBehaviour **BEHAVIOUR DEFINED AS**"

In this mapping of the accessControlDomain managed relationship class, the accessControlDomainObject class (a subclass of the accessControlRules class) participates in the adfRole and the notificationEmitter class participates in the aefRole; any managed object may participate in the memberObjectRole. The memberObjectAttribute in the accessControlDomainObject identifies the participants in the memberObjectRole and the notificationEmitter-accessControlRules name binding contains the aef participant within the adf participant.

The QUERY queryAccessControlDomain relationship management operation maps to two operations, namely :

- (a) a GET of the memberObjectAttribute of the object fulfilling the adfRole; followed by
- (b) a scoped GET of the nameBinding attribute with a scope level of one on the object fulfilling the adf role to determine the contained objects that have the value of their name binding attribute equal to «ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules.";;

**ROLE** memberObjectRole **RELATED-CLASSES** "ITU-T Rec. X.721 | ISO/IEC 10165-2":top,  
**REPRESENTED-BY ATTRIBUTE** memberObjectAttribute ;

**ROLE** aefRole

**RELATED-CLASSES** "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter

**REPRESENTED-BY NAMING**

"ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules USING  
SUBORDINATE,

**ROLE** adfRole

**RELATED-CLASSES** accessControlDomainObject

**REPRESENTED-BY NAMING**

"ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules USING  
SUPERIOR,

**OPERATIONS MAPPING**

**BIND** bindMember memberObjectRole

**MAPS-TO-OPERATION** ADD memberObjectAttribute OF adfRole,

**UNBIND** unbindMember memberObjectRole

**MAPS-TO-OPERATION** REMOVE memberObjectAttribute OF adfRole,

**QUERY** queryAccessControlDomain memberObjectRole

**MAPS-TO-OPERATION** GET memberObjectAttribute OF adfRole

**MAPS-TO-OPERATION** GET

"CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF adfRole;

**REGISTERED AS** {GRMExample.grmEx-RelationshipMapping simpleAccessControlDomainArc(1) };

**accessControlDomainObject** MANAGED OBJECT CLASS  
 DERIVED FROM "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules;  
 CHARACTERIZED BY accessControlDomainPackage PACKAGE  
 BEHAVIOUR accessControlDomainBehaviour  
 BEHAVIOUR DEFINED AS "

Membership of the access control domain is identified and modified by operations upon the memberObjectAttribute.";;

ATTRIBUTES memberObjectAttribute GET-REPLACE ADD-REMOVE;;  
 REGISTERED AS { GRMExample.grmEx-Object accessControlDomainObjectArc(1) };

## F.5.2 Access control domain relationship representation using a relationship object

**coordinatedAccessControlDomain** RELATIONSHIP MAPPING  
 RELATIONSHIP CLASS accessControlDomain;  
 BEHAVIOUR coordinatedAccessControlDomainBehaviour  
 BEHAVIOUR DEFINED AS"

In this mapping of the accessControlDomain managed relationship class, the accessControlRules class participates in the adfRole and the notificationEmitter class participates in the aefRole; any managed object may participate in the memberObjectRole. The relationship is represented by the accessControlDomainCoordinator, a subclass of the genericRelationshipObject, using the memberObjectAttribute, aefAttribute, and adfAttribute attributes.

The QUERY queryAccessControlDomain relationship management operation maps to three GET operations on the relationship object, namely:

- (a) a GET of the memberObjectAttribute;
- (b) a GET of the aefAttribute; and
- (c) a GET of the adfAttribute.";;

**RELATIONSHIP OBJECT** accessControlDomainCoordinator;

**ROLE** memberObjectRole  
 RELATED-CLASSES "CCITT Rec. X.721 | ISO/IEC 10165-2":top  
 REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER memberObjectAttribute,

**ROLE** aefRole  
 RELATED-CLASSES  
 "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter  
 REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER aefAttribute,

**ROLE** adfRole  
 RELATED-CLASSES "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules  
 REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER adfAttribute;

**OPERATIONS MAPPING**

**BIND** bindMember  
 MAPS-TO-OPERATION ADD memberObjectAttribute OF RELATIONSHIP OBJECT,  
**UNBIND** unbindMember  
 MAPS-TO-OPERATION REMOVE memberObjectAttribute OF RELATIONSHIP OBJECT,  
**QUERY** queryAccessControlDomain  
 MAPS-TO-OPERATION GET memberObjectAttribute OF RELATIONSHIP OBJECT  
 MAPS-TO-OPERATION GET aefAttribute OF RELATIONSHIP OBJECT  
 MAPS-TO-OPERATION GET adfAttribute OF RELATIONSHIP OBJECT;

**REGISTERED AS**  
 { GRMExample.grmEx-RelationshipMapping coordinatedAccessControlDomainArc(2)};

**accessControlDomainCoordinator** MANAGED OBJECT CLASS  
 DERIVED FROM genericRelationshipObject;  
 CHARACTERIZED BY accessControlDomainCoordinatorPackage PACKAGE

**ATTRIBUTES**  
 memberObjectAttribute  
 ATTRIBUTE DERIVED FROM participantPointer;  
 REGISTERED AS { GRMExample.grmEx-Attribute memberObjectAttributeArc(1) };  
 GET-REPLACE ADD-REMOVE,

**aefAttribute**

**ATTRIBUTE DERIVED FROM participantPointer;**

**REGISTERED AS { GRMExample.grmEx-Attribute aefAttributeArc(1) }; GET,**

**adfAttribute**

**ATTRIBUTE DERIVED FROM participantPointer;**

**REGISTERED AS { GRMDEexample.grmEx-Attribute adfAttributeArc(1) }; GET;**

**REGISTERED AS {GRMExample.grmEx-Object accessControlDomainCoordinatorArc(1)};**

## Annexe G

### Commentaires

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

#### G.1 Introduction

Les commentaires qui suivent ont été élaborés à partir de la liste de questions établie au cours de l'élaboration de la norme.

#### G.2 Dépendance entre objets gérés dans une relation gérée

- **Question:** la nature de la gestion GRM implique que les objets gérés participant à une relation gérée interagissent entre eux. Ceci est exprimé par un invariant portant sur les propriétés des participants. Comment spécifier cet invariant?
- **Commentaire:** des projets antérieurs de gestion GRM ont tenté de mettre en évidence et de définir une notation pour divers types d'invariants tels que les contraintes de valeurs d'attribut ou l'existence de dépendances. Comme il a été reconnu que les squelettes de comportement ont la potentialité d'exprimer tout type d'invariant, le comité NB a pris d'une manière systématique la décision de ne pas prendre en charge une notation pour un type d'invariant particulier. Il s'ensuit que tous les invariants sont exprimés sous la forme de comportement de relation gérée. L'invariant est spécifié sous la forme de propriétés de la relation gérée (rôles, opérations de gestion de relation, etc.). Le squelette de mappage de relation peut fournir un mappage des invariants utilisant la méthode de représentation (objets gérés participants, objets de relation, pointeurs de participant, etc.).

Les invariants sont, par définition, des exigences auxquelles doivent satisfaire des mises en œuvre conformes. La gestion GRM ne stipule pas de mécanisme général permettant de satisfaire à ces exigences, mais les squelettes de mappage de relation fournissent les outils permettant à ceux qui spécifient des relations gérées de spécifier de tels mécanismes pour des cas particuliers de mappage de relation.

#### G.3 Cohérence des vues

- **Question:** une méthode de représentation peut spécifier de l'information de gestion, par exemple des pointeurs de participant ou des objets de relation, qui ne concerne que cette méthode de représentation. Comment assurer la cohérence de cette information?
- **Commentaire:** un concept fondamental de la gestion GRM est que la sémantique de la relation gérée est exprimée d'une manière cohérente dans tous les éléments d'une mise en œuvre. Il s'ensuit que si un mappage de relation fait le choix de représenter la sémantique de participation d'un objet géré sous la forme de pointeurs homologues dans les objets participants, une mise en œuvre doit assurer que les pointeurs sont toujours en cohérence. En outre, si un mappage de relation fait le choix de représenter une opération BIND comme opération «Add» en utilisant des attributs pour l'un des membres d'un couple de pointeurs de participants homologues, la mise en œuvre a l'exigence d'ajuster le second pointeur afin de maintenir la cohérence. La gestion GRM ne spécifie que des exigences de cohérence de l'information mais non les mécanismes à utiliser pour maintenir cette cohérence, que ce soit au sein d'un système géré unique ou entre de multiples systèmes gérés.

#### G.4 Expression d'opérations et de notifications de gestion de relation

- **Question:** comment exprimer des opérations et notifications de gestion de relation et comment sont-elles mappées avec les opérations de gestion-systèmes?
- **Commentaire:** les opérations et notifications de gestion de relation sont exprimées sous la forme d'un certain nombre de prototypes d'opérations et de notifications qui sont ensuite mappées avec des opérations et notifications de gestion-systèmes. Le texte final fournit les détails complets de cette technique et en donne des exemples.

## G.5 Gestion générique

- **Question:** est-il possible de définir des mécanismes permettant la gestion d'un large éventail de types de relation gérée?
- **Commentaire:** une norme reliée, la Fonction générale de gestion de relations, a été définie en parallèle à la gestion GRM. Toutefois, une investigation complémentaire a indiqué que des outils génériques de gestion de relations ne seraient que d'une utilité limitée compte tenu du large éventail de types de relations qui pourraient être définies. Il a été décidé en conséquence de fournir aux personnes spécifiant des relations gérées les outils permettant de spécifier de tels mécanismes sur la base de chaque relation individuelle. La gestion GRM définit un squelette pour le mappage d'opérations et de notifications de relation gérée ainsi qu'une information générique de gestion.

Compte tenu du fait que les sous-classes de relation gérée sont compatibles avec leurs hyperclasses, une gestion générique est fournie au moins par la hiérarchie d'héritage.

## G.6 Connaissance des relations

- **Question:** comment un objet géré «sait-il» qu'il est impliqué dans une relation gérée?
- **Commentaire:** une vue anthropomorphique d'une relation n'est d'aucune utilité. Un objet géré doit satisfaire aux exigences du rôle telles que celles qui sont modélisées par la relation gérée. Une mise en œuvre doit assurer en fin de compte que la sémantique de la relation est respectée et que les mises en œuvre des objets gérés satisfont aux exigences du rôle.

## G.7 Spécification de rôle

- **Question:** un rôle devrait-il être spécifié d'une manière séparée?
- **Commentaire:** les rôles ont été envisagés au départ comme des spécifications indépendantes et réutilisables. Les réflexions ultérieures ont indiqué que les rôles sont intimement liés avec leurs relations gérées, de sorte que des spécifications faites à part n'ont qu'une valeur limitée.

## G.8 Réutilisation de spécifications

- **Question:** la réutilisation de spécifications est un aspect important de la gestion-systèmes OSI, comment est-elle mise en œuvre dans la gestion GRM?
- **Commentaire:** les sous-classes des classes de relation gérée sont *consistantes* avec leurs types supérieurs dans le sens qu'une instance d'une sous-classe peut être substituée à une instance d'une hyperclasse sans affecter l'exploitation du système géré. Les sous-classes sont en fait des types subordonnés dans la définition de traitement ODP du terme. Il s'ensuit que les outils d'héritage et de spécialisation fournissent un mécanisme de réutilisation de spécifications.

## G.9 AND SUBCLASSES

- **Question:** la déclaration AND SUBCLASSES n'a pas été reprise de la définition GDMO du modèle **name-binding** pour définir l'élément **role-mapping-specification** qui intervient dans la production du modèle **RELATIONSHIP MAPPING**.
- **Commentaire:** la capacité pour une sous-classe de prendre en charge un rôle est considérée comme une propriété fondamentale pour un objet géré, et doit donc être inconditionnellement héritée.

## G.10 Relations entre relations

- **Question:** comment est-il possible de modéliser des relations entre relations?
- **Commentaire:** comme la gestion GRM modélise des relations entre objets gérés, si des relations sont représentées sous la forme d'objets de relation, il n'y a aucune raison que la gestion GRM ne puisse modéliser des relations de relations. La gestion GRM ne fournit à cet effet aucune prise en charge particulière, mais le squelette BEHAVIOUR permet de modéliser toute sémantique complémentaire.

**G.11 Espace de noms des objets de relation**

- **Question:** quel devrait être le domaine de validité des noms des objets de relation?
- **Commentaire:** il y a eu des discussions au sujet de l'attribution de noms de l'ensemble des objets de relation d'un système géré au sein du domaine de validité d'un objet unique appartenant à une classe particulière, souvent appelée classe d'*objets d'ancrage* avec le but d'accéder à l'ensemble des objets de relation d'un système géré au moyen de l'accès au domaine CMIS. La conclusion a été que, comme les normes de gestion en vigueur considèrent la structure de nomenclature comme un problème local, il aurait été incompatible pour la gestion GRM de spécifier une structure particulière.

**G.12 Méthodes de représentation permises**

- **Question:** les méthodes de représentation peuvent-elles représenter tous les types de relations?
- **Commentaire:** non; certaines méthodes de représentation sont restreintes de manière inhérente au type de relations qu'elles peuvent représenter. Le Tableau G.1 indique les types de relations pouvant être représentées par les diverses méthodes.

**Tableau G.1 – Méthodes de représentation permises**

Méthode de représentation	Cardinalité de relation = 1			Cardinalité de relation > 1		
	Cardinalité de rôle			Cardinalité de rôle		
	1:n	n:m	n:m:p	1:n	n:m	n:m:p
Dénomination	oui	non	non	non	non	non
Pointeurs de participants	oui	oui	oui	non	non	non
Objet de relation	oui	oui	oui	oui	oui	oui
Opérations de gestion des systèmes	oui	oui	oui	oui	oui	oui