



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

G.851.1

(11/96)

SÉRIE G: SYSTÈMES ET SUPPORTS DE
TRANSMISSION, SYSTÈMES ET RÉSEAUX
NUMÉRIQUES

Systemes de transmission numériques – Réseaux
numériques – Réseau de gestion des télécommunications

**Gestion du réseau de transport – Application du
modèle de référence RM-ODP**

Recommandation UIT-T G.851.1

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE G
SYSTÈMES ET SUPPORTS DE TRANSMISSION, SYSTÈMES ET RÉSEAUX NUMÉRIQUES

CONNEXIONS ET CIRCUITS TÉLÉPHONIQUES INTERNATIONAUX	G.100–G.199
SYSTÈMES INTERNATIONAUX ANALOGIQUES À COURANTS PORTEURS	
CARACTÉRISTIQUES GÉNÉRALES COMMUNES À TOUS LES SYSTÈMES ANALOGIQUES À COURANTS PORTEURS	G.200–G.299
CARACTÉRISTIQUES INDIVIDUELLES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX À COURANTS PORTEURS SUR LIGNES MÉTALLIQUES	G.300–G.399
CARACTÉRISTIQUES GÉNÉRALES DES SYSTÈMES TÉLÉPHONIQUES INTERNATIONAUX HERTZIENS OU À SATELLITES ET INTERCONNEXION AVEC LES SYSTÈMES SUR LIGNES MÉTALLIQUES	G.400–G.449
COORDINATION DE LA RADIODÉLÉPHONIE ET DE LA TÉLÉPHONIE SUR LIGNES	G.450–G.499
CARACTÉRISTIQUES DES SUPPORTS DE TRANSMISSION	G.600–G.699
SYSTÈMES DE TRANSMISSION NUMÉRIQUES	
EQUIPEMENTS TERMINAUX	G.700–G.799
Généralités	G.700–G.709
Codage des signaux analogiques en modulation par impulsions et codage	G.710–G.719
Codage des signaux analogiques par des méthodes autres que la MIC	G.720–G.729
Principales caractéristiques des équipements de multiplexage primaires	G.730–G.739
Principales caractéristiques des équipements de multiplexage de deuxième ordre	G.740–G.749
Caractéristiques principales des équipements de multiplexage d'ordre plus élevé	G.750–G.759
Caractéristiques principales des équipements de transcodage et de multiplication numérique	G.760–G.769
Fonctionnalités de gestion, d'exploitation et de maintenance des équipements de transmission	G.770–G.779
Caractéristiques principales des équipements de multiplexage en hiérarchie numérique synchrone	G.780–G.789
Autres équipements terminaux	G.790–G.799
RÉSEAUX NUMÉRIQUES	G.800–G.899
Généralités	G.800–G.809
Objectifs de conception pour les réseaux numériques	G.810–G.819
Objectifs de qualité et de disponibilité	G.820–G.829
Fonctions et capacités du réseau	G.830–G.839
Caractéristiques des réseaux à hiérarchie numérique synchrone	G.840–G.849
Réseau de gestion des télécommunications	G.850–G.859
SECTION NUMÉRIQUES ET SYSTÈMES DE LIGNES NUMÉRIQUES	G.900–G.999
Généralités	G.900–G.909
Paramètres pour les systèmes à câbles optiques	G.910–G.919
Sections numériques à débits hiérarchisés multiples de 2048 kbit/s	G.920–G.929
Systèmes numériques de transmission par ligne à débits non hiérarchisés	G.930–G.939
Systèmes de transmission numérique par ligne à supports MRF	G.940–G.949
Systèmes numériques de transmission par ligne	G.950–G.959
Section numérique et systèmes de transmission numérique pour l'accès usager du RNIS	G.960–G.969
Systèmes sous-marins à câbles optiques	G.970–G.979
Systèmes de transmission par ligne optique pour les réseaux locaux et les réseaux d'accès	G.980–G.999

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

RECOMMANDATION UIT-T G.851.1

GESTION DU RESEAU DE TRANSPORT – APPLICATION DU MODELE DE REFERENCE RM-ODP

Source

La Recommandation UIT-T G.851.1, révisée par la Commission d'études 15 de l'UIT-T (1993-1996), a été approuvée le 8 novembre 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

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

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

Dans certains secteurs 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.

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© 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.

TABLE DES MATIÈRES

		Page
1	Domaine d'application.....	1
1.1	Objectifs.....	1
1.2	Principes.....	1
1.3	Structure de la présente Recommandation.....	2
2	Références normatives.....	2
3	Définitions	3
4	Abréviations utilisées dans la Recommandation G.851-1	3
5	Aperçu méthodologique.....	4
5.1	Introduction.....	4
5.1.1	Descriptions des perspectives du modèle RM-ODP.....	5
5.2	Utilisation des perspectives du modèle RM-ODP dans la spécification d'une méthodologie de conception	6
5.3	Fonction de mise en correspondance des perspectives.....	6
5.3.1	Etiquetage	6
6	Perspective entreprise	7
6.1	Domaine d'application de la perspective entreprise.....	8
6.2	Concepts.....	8
6.2.1	Communauté.....	8
6.2.2	Contrat	8
6.2.3	Rôles	9
6.2.4	Règles de gestion.....	9
6.2.5	Actions.....	10
6.2.6	Activités.....	10
6.2.7	Services.....	10
6.2.8	Caractéristique de service.....	10
6.2.9	Combinaison des services.....	11
6.3	Extension de communautés	11
6.4	Domaines	11
7	Perspective informationnelle	11
7.1	Perspective informationnelle	11
8	Perspective informatique	12
8.1	Concepts informatiques	13
8.1.1	Perspective informatique indépendante du domaine de communication	13
8.1.2	Mappages vers la spécification de la perspective informatique dépendante du domaine de communication.....	14

	Page
9	Perspective d'ingénierie 15
9.1	Introduction..... 15
9.2	Concepts d'ingénierie..... 15
9.3	Perspective d'ingénierie basée sur la gestion OSI..... 15
9.3.1	Influence des protocoles sur les perspectives informationnelle et informatique..... 16
9.3.2	Contraintes d'ingénierie imposées par la perspective entreprise 17
9.3.3	Utilisation de paquetages et d'objets gérés GDMO 17
9.3.4	Prise en charge de services de gestion multiples au niveau de l'interface d'ingénierie..... 17
9.3.5	Identification..... 18
9.3.6	Mappage de relation 18
9.3.7	Comparaison entre l'utilisation d'actions par rapport à des opérations d'attribut pour des modifications de relations et d'état 18
9.4	L'environnement de traitement réparti..... 18
9.4.1	Aperçu général..... 18
10	Utilisation d'ensembles dans la définition d'applications de gestion de réseau 19
10.1	Domaine d'application 19
Annexe A – Squelette et directives générales pour la spécification de la perspective entreprise..... 20	
A.1	Définition non formelle du squelette entreprise..... 20
A.2	Définition formelle du squelette entreprise..... 21
Annexe B – Structure de la perspective informationnelle..... 23	
B.1	Introduction..... 23
B.2	Descriptions du modèle 24
B.3	Structure de la spécification..... 24
B.3.1	Classes d'objets d'information 24
B.3.2	Relations entre informations..... 26
B.3.3	Définition du schéma statique 28
B.3.4	Schémas dynamiques..... 29
B.3.5	Attributs 30
Annexe C – Description du squelette de la perspective informatique 31	
C.1	Introduction..... 31
C.2	Directives générales..... 32
C.2.1	Directives générales propres au squelette d'opération..... 32
C.2.2	Définitions d'interface client serveur..... 35
C.2.3	Mappage vers divers domaines de communication..... 36

	Page
C.3 Définitions formelles de squelette	36
C.3.1 Squelette de classe d'objets.....	36
C.3.2 Squelette d'interface.....	37
C.3.3 Squelette d'opération.....	38
C.4 Exemple	39
Annexe D – Squelettes et directives générales de la perspective d'ingénierie de la gestion OSI.....	42
D.1 Squelettes	42
D.2 Mappages possibles vers les définitions d'objets gérés.....	42
Annexe E – Syntaxe d'étiquette.....	42
E.0 Introduction.....	42
E.1 Définition de la syntaxe d'étiquette en notation BNF.....	43
E.2 Structure d'arborescence d'étiquette dans la perspective entreprise.....	44
E.2.1 Exemples d'utilisation.....	45
E.3 Structure de l'arborescence d'étiquette de la perspective informationnelle.....	45
E.3.1 Exemples d'utilisation.....	46
E.4 Structure de l'arborescence d'étiquette dans la perspective informatique.....	46
E.4.1 Exemples d'utilisation.....	47
Annexe F – Squelette ensemble du modèle Ensemble.....	47
F.1 La technique du modèle Ensemble	47
F.2 Squelette du modèle Ensemble	47
F.2.1 Introduction.....	48
F.2.2 Contexte de gestion	48
F.2.3 Modèle d'information de gestion.....	48
F.2.4 Exigences de conformité du modèle Ensemble.....	48
Appendice I – Exemples de squelettes et de directives de spécification.....	48
I.1 Comparaison entre les services fournis à l'entreprise et les contrats	48
Appendice II – Représentation d'états combinés.....	49
Appendice III – Description de la réalisation du service.....	51
III.1 Domaine d'application	51
III.2 Concepts.....	52
Appendice IV – Exemple d'utilisation du concept et du format du modèle Ensemble.....	54
Appendice V – Exemple de processus d'élaboration de spécification.....	57

	Page
Appendice VI – Mappage entre perspectives.....	58
VI.1 Démarche	58
VI.2 Mappages de la perspective informationnelle.....	59
VI.2.1 Objets et relations d'information.....	59
VI.2.2 Schéma statique	59
VI.2.3 Schéma dynamique.....	60
VI.3 Mappages de la perspective informatique.....	60
VI.3.2 Interface informatique.....	60
VI.3.3 Objet informatique.....	60
Appendice VII – Directives pour l'utilisation de la notation Z dans la perspective informationnelle.....	60
VII.1 Introduction.....	60
VII.2 Présentation de la notation Z.....	61
VII.2.1 Schémas	61
VII.2.2 Symboles.....	61
VII.2.3 Exemple	62
VII.3 Conventions de spécification	63
VII.3.1 Spécification d'attribut.....	63
VII.3.2 Spécification d'objet	63
VII.3.3 Spécification de relation	64

Recommandation G.851.1

GESTION DU RESEAU DE TRANSPORT – APPLICATION DU MODELE DE REFERENCE RM-ODP

(Genève, 1996)

1 Domaine d'application

La présente Recommandation fournit une description des concepts de modélisation sur lesquels se base le modèle de niveau réseau de la Commission d'études 15 de l'UIT-T. Ce modèle utilise le cadre général du modèle de référence des traitements répartis ouverts (RM-ODP, *reference model of open distributed processing*) [1], [2], [3] et [4] comme point de départ pour la définition d'une méthodologie prescrivant la démarche de définition du modèle au niveau réseau.

L'Appendice V décrit un processus d'élaboration d'un ensemble complet de spécifications basées sur la présente méthodologie.

1.1 Objectifs

La présente Recommandation vise les objectifs suivants:

- définir une méthodologie de spécification d'un modèle de gestion de réseau pour la prise en charge des interfaces vers un système d'exploitation réseau au sein de l'architecture RGT, utilisant au départ la gestion OSI et pouvant utiliser d'autres infrastructures lorsque les normes correspondantes seront disponibles (par exemple le langage IDL CORBA et les fonctions ODP);
- produire des modèles qui peuvent être répartis d'une manière souple entre différentes architectures de système de gestion (par exemple, au moyen de la gestion OSI ou au moyen de systèmes faisant appel à des traitements répartis);
- l'ouvrage fournira une possibilité maximale de réutilisation des spécifications et des entités de traitement;
- cet ouvrage conservera une compatibilité maximale avec les Recommandations existantes fondées sur la gestion OSI et doit être considéré comme une extension de la base installée.

NOTE – La présente méthodologie définit des perspectives entreprise, informationnelle et informatique indépendantes de l'implémentation. De cette manière, les définitions utilisées par exemple pour la gestion OSI, peuvent également s'appliquer à des définitions ultérieures utilisant d'autres infrastructures. En conséquence, alors que les premières Recommandations utilisant la présente méthodologie se concentreront sur l'infrastructure de la gestion OSI, les modèles définis dans ces Recommandations pour les perspectives entreprise, informationnelle et informatique sera indépendant de l'infrastructure.

1.2 Principes

Le modèle RM-ODP a été choisi afin de fournir un cadre général pour une technique de spécification rigoureuse dont la trace peut être suivie jusqu'au niveau de la définition des besoins.

Le choix des techniques est guidé par l'objectif de maintenir une compatibilité maximale avec la gestion OSI.

Maintenir la compatibilité avec les Recommandations existantes basées sur la gestion OSI (M.3100, G.774-X, série X.700 sur la gestion OSI, Q.821, Q.822, etc).

Assurer l'équilibre entre les exigences de lisibilité par un utilisateur humain, la lisibilité par la machine et la précision (utilisation de notations formelles, par exemple).

La présente Recommandation doit faciliter l'interfonctionnement d'applications basées sur des implémentations utilisant des infrastructures différentes.

1.3 Structure de la présente Recommandation

Le paragraphe 2 fournit une liste des références utilisées par la présente Recommandation, le paragraphe 3 contient les définitions qui ont été introduites par la présente Recommandation et le paragraphe 4 définit les abréviations utilisées. Le paragraphe 5 donne un aperçu général de la méthodologie décrite dans la présente Recommandation. Les paragraphes 6, 7, 8 et 9 donnent les détails spécifiques des perspectives du modèle RM-ODP utilisé par la méthodologie, qui sont respectivement les perspectives entreprise, informationnelle, informatique et d'ingénierie. L'article 10 décrit l'utilisation du modèle Ensemble dans la définition d'applications de gestion de réseau.

Les Annexes A, B, C et D contiennent les descriptions des squelettes utilisés pour définir respectivement les perspectives entreprise, informationnelle, informatique et d'ingénierie. L'Annexe E définit la syntaxe des étiquettes utilisées dans la méthodologie, et l'Annexe F décrit les squelettes utilisés pour définir un modèle ensemble de gestion.

Les Appendices I à V contiennent divers exemples de l'application de la méthodologie à la résolution de problèmes de gestion de réseau et les manières d'utiliser la méthodologie pour l'élaboration de spécifications de normes. L'Appendice VI définit le mappage entre les diverses perspectives du modèle RM-ODP utilisées par la méthodologie. L'Appendice VII fournit quelques directives pour l'utilisation de la notation Z dans la perspective informationnelle.

2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui de ce fait en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

- [1] Recommandation UIT-T X.901¹ | ISO/CEI 10746-1¹, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: aperçu général et guide d'utilisation.*
- [2] Recommandation UIT-T X.902 (1995) | ISO/CEI 10746-2:1996, *Technologies de l'information – Traitement ouvert réparti – Modèle de référence: fondements.*
- [3] Recommandation UIT-T X.903 (1995) | ISO/CEI 10746-3:1996, *Technologies de l'information – Traitement ouvert réparti – Modèle de référence: architecture.*
- [4] Recommandation UIT-T X.904¹ | ISO/CEI 10746-4¹, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: sémantique architecturale.*
- [5] 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.*
- [6] Recommandation UIT-T X.725 (1995) | ISO/CEI 10165-7:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: modèle général de relation.*

¹ Actuellement à l'état de projet.

- [7] SPIVEY (J.M.): The Z Notation – A Reference Manual, 2nd Edition, *Prentice Hall International*, ISBN 0-13-978529-9, 1992.
- [8] NM Forum 025, OMNIPoint 1: The "Ensemble" Concepts and Format, août 1992.
- [9] Recommandation UIT-T X.724 (1996) | ISO/CEI 10165-6:1997, *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'implémentations associés à la gestion OSI.*
- [10] Recommandation UIT-T G.852.1 (1996), *Gestion du réseau de transport – Point de vue Entreprise pour la gestion de connexions de sous-réseau simple.*
- [11] Recommandation UIT-T G.774 (1996), *Modèle d'information de gestion de la hiérarchie numérique synchrone du point de vue des éléments de réseau.*
- [12] Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2:1996, *Technologies de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*

3 Définitions

La présente Recommandation définit les termes suivants:

3.1 type de contrat: expression de toutes les caractéristiques du contrat (obligatoires, négociables ou optionnelles).

3.2 instance de contrat: résultat d'une négociation entre un fournisseur et un client donné.

4 Abréviations utilisées dans la Recommandation G.851-1

La présente Recommandation utilise les abréviations suivantes:

ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
BEO	objet d'ingénierie de base (<i>basic engineering object</i>)
BNF	formalisme de Backus-Naur (<i>backus-naur form</i>)
CMIP	protocole commun d'information de gestion (<i>common management information protocol</i>)
CMISE	élément commun de service d'information de gestion (<i>common management information service element</i>)
CORBA	architecture commune de demande de courtage d'objets (<i>common object request broker architecture</i>)
DCE	environnement informatique réparti (<i>distributed computing environment</i>)
DPE	environnement de traitement réparti (<i>distributed processing environment</i>)
GDIO	directive pour la définition des objets d'information (<i>guideline for the definition of information objects</i>)
GDMO	directives pour la définition des objets gérés (<i>guidelines for the definition of managed objects</i>)
GRM	modèle général de relation (<i>general relationship model</i>)
IDL	langage de définition d'interface (<i>interface definition language</i>)
MOCS	déclaration de conformité d'objet géré (<i>managed object conformance statement</i>)
ODL	langage de définition d'objets (<i>object definition language</i>)

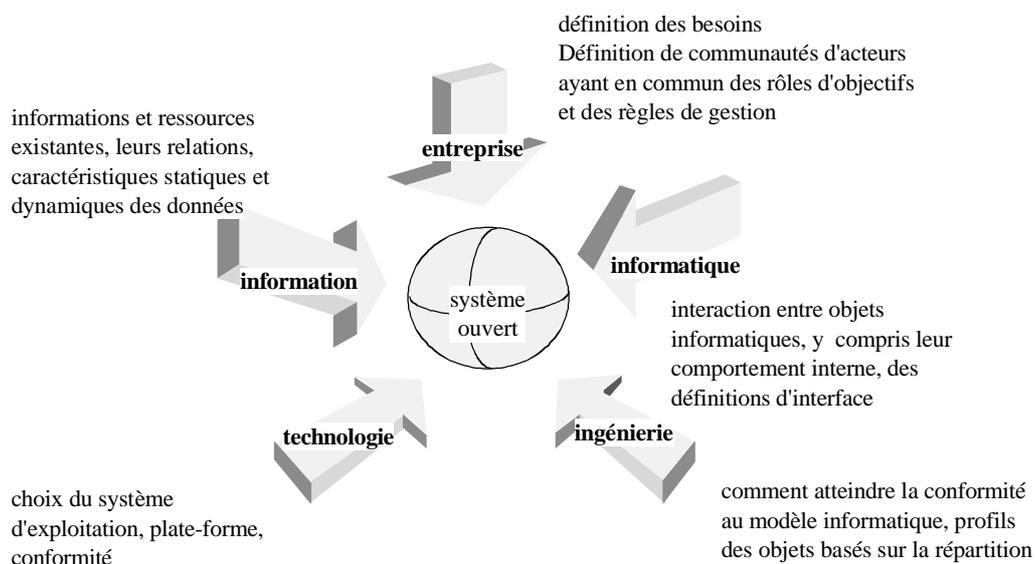
ODP	traitement réparti ouvert (<i>open distributed processing</i>)
OMG	groupe de gestion d'objets (<i>object management group</i>)
OSI	interconnexion des systèmes ouverts (<i>open systems interconnection</i>)
PICS	déclaration de conformité d'implémentation de protocole (<i>protocol implementation conformance statement</i>)
RGT	réseau de gestion de télécommunications
SDH	hiérarchie numérique synchrone (<i>synchronous digital hierarchy</i>)
SMI	information de gestion-systèmes (<i>systems management information</i>)
SNC	connexion de sous-réseau (<i>subnetwork connection</i>)
SNMP	protocole simple de gestion de réseau (<i>simple network management protocol</i>)
UIT-T	secteur de normalisation des télécommunications de l'Union internationale des télécommunications

5 Aperçu méthodologique

5.1 Introduction

Le modèle RM-ODP fournit, du point de vue de la gestion de réseau, un cadre général orienté objets (pour des systèmes répartis de gestion) qui permet de définir les prescriptions de l'utilisateur pour chaque application de gestion (par exemple la gestion de configuration), les informations (ou données) liées aux ressources gérées et les modes d'accès aux données et de manipulation de celles-ci, ces définitions étant données d'une manière intrinsèquement indépendante de la technologie et du mode de répartition utilisé pour l'implémentation du système de gestion.

Comme illustré par la Figure 1, le cadre général du modèle RM-ODP fournit cinq perspectives du système et des ressources gérés.



T1521040-96

Figure 1/G.851.1 – Perspectives du modèle RM-ODP

Le modèle RM-ODP fournit:

- un ensemble de concepts permettant de décrire des systèmes répartis ouverts;
- l'indépendance par rapport à toute méthode et tout langage d'analyse, de conception et de programmation.

La modélisation initiale faite au niveau réseau n'utilise que les perspectives entreprise, informationnelle, informatique et d'ingénierie du modèle RM-ODP. Il est considéré actuellement que la perspective technologique se situe en dehors du domaine du secteur de la normalisation des télécommunications de l'UIT.

Les perspectives informationnelle et informatique du modèle au niveau réseau se concentrent sur la sémantique plutôt que sur la syntaxe des informations qui doivent être échangées pour chaque application de gestion, ou pour chaque ensemble de telles applications. Ce niveau de définition est suffisant pour permettre une spécification plus détaillée dans la perspective d'ingénierie lorsque l'interopérabilité du système est définie par les interfaces externes. Les prescriptions de conformité seront définies, d'une manière générale, par rapport à ces interfaces externes. La perspective d'ingénierie sera définie pour une technologie de communication donnée, telle que la gestion-système OSI.

Le présent document méthodologique définit les perspectives utilisées par la modélisation au niveau réseau.

5.1.1 Descriptions des perspectives du modèle RM-ODP

Des descriptions complètes des perspectives du modèle RM-ODP se trouvent dans les références énumérés au paragraphe 2. Une brève description des perspectives du modèle RM-ODP est donnée ci-après:

- perspective entreprise: une vue d'un système ODP et de son environnement qui se concentre sur les buts, le domaine d'application et les règles de gestion du système. Une spécification entreprise permet au client d'un système ODP d'exprimer ses besoins et ses règles de gestion, créant de ce fait un contrat entre le client et le fournisseur du système ODP. Il s'ensuit qu'une spécification entreprise doit être exprimée dans des termes compréhensibles pour les deux parties;
- perspective informationnelle: une vue d'un système ODP et de son environnement qui se concentre sur la sémantique des informations et des activités de traitement de l'information du système. Une perspective informationnelle se concentre sur la définition des types d'objets d'information, de leurs relations, de leurs valeurs d'état et des modifications d'état autorisées;
- perspective informatique: une vue d'un système ODP et de son environnement qui se concentre sur la décomposition fonctionnelle en structures se prêtant à une répartition. Une spécification informatique décrit des types d'objets informatiques et leurs types d'interfaces. Des instances de ces types d'objets coopéreront à travers des interfaces. Les types d'interfaces d'opération sont définis par la spécification de leurs signatures d'opération et par les comportements correspondants. Tous les objets informatiques sont potentiellement répartis au sein de systèmes distincts. Dans le contexte de la présente méthodologie, tout objet informatique décrit ne peut être réparti d'une manière supplémentaire que si de nouvelles interfaces informatiques sont définies;

- perspective d'ingénierie: une vue d'un système ODP et de son environnement qui se concentre sur les fonctions nécessaires pour la prise en charge de la répartition dans ce système. Une spécification d'ingénierie est utilisée pour prendre des décisions concrètes concernant la répartition des objets informatiques et pour déterminer en outre les composants d'infrastructure nécessaires à la prise en charge des objets répartis;
- perspective technologique: une vue d'un système ODP et de son environnement qui se concentre sur le choix de la technologie du système.

5.2 Utilisation des perspectives du modèle RM-ODP dans la spécification d'une méthodologie de conception

Le modèle de référence pour les traitements répartis ouverts fournit une démarche de modélisation (les perspectives du modèle RM-ODP), mais ne prescrit pas de méthodologie pouvant être appliquée pour l'élaboration d'un système. Le présent sous-paragraphe décrit, en utilisant les perspectives du modèle RM-ODP, l'utilisation de cette méthodologie à des fins de normalisation de modèles de gestion de réseau de transport. Dans ce processus, les composants applicatifs sont considérés comme des objets communicant à travers des interfaces bien définies et pour lesquelles n'est décrit, d'une manière indépendante de l'implémentation, que le comportement pouvant être observé de l'extérieur. La méthodologie utilise les étapes suivantes:

- a) définition des besoins;
- b) définition des informations décrivant le système;
- c) description des processus de manipulation de l'information et de fourniture des services;
- d) prise des décisions de répartition et d'implémentation.

Chacune de ces étapes conception correspond à une perspective du modèle RM-ODP. Il n'est pas nécessaire, dans la pratique, que la succession chronologique de ces étapes soit rigoureusement respectée, plus importante est la démarcation entre centres d'intérêt que fournit le cadre général du modèle RM-ODP. L'Appendice VI présente un exemple de processus de spécification basé sur cette méthodologie, alors que l'Appendice VII présente les relations entre les perspectives.

L'ensemble de ressources constituant une architecture de réseau de transport peut être décrit en n'utilisant que les perspectives entreprise et information, alors que la description d'un service de gestion de réseau nécessite en plus l'utilisation de la perspective informatique. L'architecture de réseau de transport et la définition de service seront fusionnées au cours du processus d'ingénierie afin d'élaborer des fonctions de gestion, comme par exemple une interface Q3.

5.3 Fonction de mise en correspondance des perspectives

Cette méthodologie permet le suivi de la correspondance entre la définition des besoins et les spécifications d'ingénierie qui en résultent. Ceci est réalisé en structurant la documentation d'une manière mettant en évidence les relations directes entre les perspectives de chaque communauté (groupe d'applications de gestion) de l'entreprise. Par exemple, le suivi des spécifications d'ingénierie remontant vers la définition des besoins est également assuré par l'utilisation du schéma d'étiquetage défini ci-dessous.

5.3.1 Etiquetage

Une étiquette sera attribuée à des éléments particuliers d'une spécification afin de permettre à d'autres spécifications d'y faire référence. Cette référence peut être faite dans la même perspective, entre plusieurs perspectives d'une même spécification de service ou entre les spécifications de plusieurs services (par exemple en utilisant une clause IMPORT).

5.3.1.1 Structure de déclaration d'étiquette

La structure des étiquettes se base sur la notation BNF.

La référence d'étiquette possède une structure arborescente. Tout noeud de l'arbre sera soit un mot clé prédéfini (comme défini dans la présente Recommandation, par exemple ATTRIBUTE, COMMUNITY ou ROLE), soit une étiquette d'élément identifiant un élément de spécification formelle, comme par exemple networkTTP ou operational_state. Lorsqu'il est défini, l'élément d'étiquette doit être non ambigu dans le contexte du noeud supérieur. Lorsqu'elle est utilisée comme référence, une étiquette doit être non ambiguë dans le contexte. La référence de l'étiquette doit correspondre au niveau qui fournit un pointeur non ambigu vers la partie de la spécification à laquelle il doit être fait référence. Il n'est pas nécessaire, par exemple, de fournir l'élément Recommandation de l'étiquette pour des références d'étiquette au sein d'une même Recommandation ou de fournir, au sein d'une spécification de perspective, l'étiquette d'élément perspective pour des références à d'autres éléments de la même perspective. Une étiquette d'élément est définie sous la forme d'une chaîne de caractères, la référence d'étiquette étant encadrée par les signes < et >.

L'Annexe E donne la définition de la structure de l'étiquette en notation BNF.

6 Perspective entreprise

La perspective entreprise, définie dans troisième partie du modèle RM-ODP (Rec. UIT-T X.903|ISO/CEI 10746-3 [3]), indique le but, le domaine d'application et les règles de gestion d'un système ODP. Elle a pour objectif de spécifier des besoins définis du point de vue de l'ensemble des acteurs du système. Ces besoins sont exprimés sous forme d'interactions entre le système et l'environnement utilisateur. Il est nécessaire de décrire dans la perspective entreprise toute interaction entre applications de gestion qui a un impact sur le comportement du système.

La perspective entreprise se trouve dans le modèle au niveau réseau afin de:

- documenter l'utilisation des diverses parties du modèle basée en relation avec des **communautés** de fonctions communes (applications);
- fournir un moyen de spécifier les définitions de besoins à partir desquelles est défini le modèle.

Les squelettes et le langage utilisés dans la présente Recommandation sont conformes aux prescriptions définies dans la troisième partie du modèle RM-ODP [3].

Les squelettes sont utilisés pour définir des **communautés**, représentant un ensemble de fonctions liées qui ont pour but d'atteindre un objectif spécifique (application de gestion) tel que la gestion des connexions. La communauté comprend un ensemble de rôles, un ensemble d'actions et un ensemble de règles de gestion dont le but est de réaliser l'objectif de coopération, ou contrat, qui est partagé entre les rôles. Une communauté ne spécifie pas d'objets, mais uniquement les rôles qu'ils jouent. Il s'ensuit qu'un objet donné peut jouer un certain nombre de rôles différents dans des communautés différentes. Une communauté ne spécifie pas de fonctions de transparence. Une communauté prend en charge un certain nombre d'instances distinctes du contrat de communauté entre rôles. Chaque rôle (ou ensemble de rôles) possède un contrat individuel avec un autre rôle (ou ensemble de rôles). Cette instance du contrat sera une sélection particulière de caractéristiques de service à partir de l'ensemble disponible.

Les entités suivantes sont associées à chaque description de communauté:

- contrat;
- rôles de communauté;
- règles de gestion de communauté;

- actions de communauté;
- activités de communauté;
- service;
- caractéristique de service.

Les sous-paragraphe ci-après décrivent plus en détail les composants entreprise. Les squelettes et le langage permettant de les définir sont décrits dans l'Annexe A.

6.1 Domaine d'application de la perspective entreprise

Une relation entre client et fournisseur est créée par l'établissement d'un contrat. Un contrat donné exprime un service et une qualité associée qui sont offerts par le fournisseur à son client et acceptés par ce dernier.

Un contrat de service inclura toutes les caractéristiques qui établissent la définition du type de service. Il contient:

- une définition de la communauté;
- la liste des rôles impliqués dans la communauté;
- les règles de gestion s'appliquant à la communauté;
- chaque action et les règles de gestion associées;
- chaque activité éventuelle.

6.2 Concepts

6.2.1 Communauté

L'expression des besoins est faite au départ en identifiant les communautés du système ODP, une communauté étant un groupe de rôles qui se joignent en vue d'un certain objectif. L'objectif de la communauté doit être énoncé d'une manière explicite. En général, l'objectif de la communauté est la fourniture d'un certain service, par exemple une communauté de gestion de connexion de sous-réseau, une communauté de gestion de ressource, etc.

La troisième partie du modèle RM-ODP [3] donne la définition suivante:

"communauté: une combinaison d'objets formée dans le but d'atteindre un objectif. L'objectif est exprimé sous la forme d'un contrat qui spécifie de quelle manière les objectifs peuvent être atteints."

La communauté est définie par son but (c'est-à-dire l'objectif commun des rôles impliqués dans la communauté), la définition de chaque rôle impliqué dans la communauté et les règles de gestion qui s'appliquent à la communauté dans son ensemble.

6.2.2 Contrat

Le résultat de la négociation d'un service est une instance de contrat qui exprime un accord au sujet de la sélection faite au sein d'un ensemble de caractéristiques du fournisseur. Ces caractéristiques de service peuvent être fixées sous la forme d'un ensemble d'actions et d'activités prises en charge ainsi que, si cela s'applique, de règles de gestion prises en charge.

La deuxième partie du modèle RM-ODP [2] donne la définition suivante d'un contrat:

"contrat: accord régissant une partie du comportement collectif d'un ensemble d'objets. Un contrat spécifie des obligations, autorisations et interdictions qui s'appliquent aux objets concernés. La spécification d'un contrat peut inclure:

- a) une spécification des différents rôles que les objets impliqués dans le contrat peuvent assumer, ainsi que des interfaces associées à ces rôles;
- b) des attributs de qualité de service;
- c) des indications de durée ou des périodes de validité;
- d) des indications de comportement invalidant le contrat;
- e) la durée de vie et les conditions de sécurité."

Certaines caractéristiques de contrat seront requises pour tous les clients d'un fournisseur donné (ce qui impose des contraintes au comportement du fournisseur et/ou du client), d'autres seront négociées entre le fournisseur et chaque client, le résultat de la négociation étant la définition des caractéristiques prises en charge. Certaines caractéristiques du contrat peuvent rester optionnelles après l'établissement du contrat et être utilisées par le client et/ou le fournisseur sur une base négociée ou de règles de gestion locales (tel que le "meilleur effort").

L'expression de ces règles de gestion est une spécification entreprise qui sera constituée de deux parties. La première exprimera toutes les caractéristiques du contrat (obligatoires, négociables ou optionnelles). Cette première partie sera utilisée par le fournisseur pour établir la deuxième partie qui résulte de la négociation entre le fournisseur et un client donné. La première partie peut être considérée comme un "type de contrat de service" et la deuxième partie comme une "instance de contrat de service".

Le processus de négociation de contrat est en dehors du domaine d'application de la méthodologie.

NOTE – La présente méthodologie ne fait pas de distinction entre interfaces et rôles dans la perspective entreprise.

6.2.3 Rôles

La présente méthodologie utilise le concept de rôle entreprise comme défini dans la deuxième partie du modèle RM-ODP [2]. La totalité des rôles sera définie pour chaque service.

Un rôle d'appelant représente le comportement d'un objet entreprise définissant les demandes de service d'un service donné.

Un rôle de fournisseur représente le comportement d'un objet entreprise exécutant les demandes de service d'un service donné.

Les autres rôles d'un service représentent le comportement des objets entreprise qui expriment l'implication des ressources dans le contexte de ce service.

6.2.4 Règles de gestion

Les règles de gestion de la communauté sont spécifiées comme un ensemble d'autorisations, d'obligations, d'interdictions et d'exceptions s'appliquant pour le client ou le fournisseur en ce qui concerne la communauté. La partie 2 du modèle RM-ODP donne les définitions suivantes:

"règles de gestion: ensemble de règles liées à un objectif particulier. Une règle peut s'exprimer sous forme d'obligation, de permission ou d'interdiction;

NOTE – Les règles de gestion ne sont pas toutes des contraintes, certaines représentent une attribution de compétence."

"obligation: prescription stipulant la nécessité d'un comportement particulier. Une obligation est remplie par l'occurrence du comportement prescrit;"

"permission: prescription autorisant un comportement particulier. Une permission équivaut au fait qu'il n'est pas obligatoire que le comportement ne se produise pas;"

"interdiction: prescription stipulant qu'un comportement particulier ne doit pas se manifester. Une interdiction équivaut au fait qu'il est obligatoire que le comportement ne se présente pas."

6.2.5 Actions

La partie 2 du modèle RM-ODP donne la définition suivante:

"action: quelque chose qui se produit."

Toute communauté possède un ensemble d'actions qui prend en charge le but de la communauté. Les actions seront utilisées pour exprimer les demandes de service et les réponses associées échangées entre le client et le serveur. Les actions caractéristiques sont les suivantes: établissement de la connexion, libération de la connexion et modification de la connexion. Ces actions peuvent être regroupées en activités qui spécifient l'ordre dans lequel peuvent s'exécuter les actions.

Toute action est définie par un nom d'action et la spécification des règles de gestion de l'action.

A chaque action est associé un ensemble de **règles de gestion d'action** décrites comme une obligation, une autorisation ou une interdiction s'appliquant, soit au client soit au fournisseur en ce qui concerne la communauté. Ces règles de gestion stipulent le rôle et les informations impliqués dans chaque règle de gestion. La perspective entreprise n'est toutefois pas prévue pour fournir des prescriptions concernant les informations.

6.2.6 Activités

Il peut être, défini au sein d'une communauté, un ensemble d'activités incluant des descriptions d'activités et un nom d'activité. Pour chaque activité, il peut être défini un **graphe d'actions** contenant une liste d'actions élémentaires associées à l'activité. A chaque activité est également associé un ensemble de **règles de gestion d'activité** qui, avec le graphe d'action, garantit que les interdépendances, telles que l'ordre de succession des actions au sein d'une activité sont stipulées clairement sous forme de règle de gestion.

Conformément à la définition du modèle RM-ODP, les activités font généralement partie d'une réalisation de service et l'unique échange entre un client et un fournisseur qui apparaissent dans la spécification du contrat est l'action de tête d'une activité du fournisseur. En fait, c'est sur la base de la réalisation d'un service qu'il est déterminé si l'action de tête doit être suivie ou non d'actions ultérieures. Dans certains cas toutefois, le client doit indiquer au fournisseur plusieurs actions liées comme faisant partie d'une caractéristique de service, auquel cas ces actions liées font partie d'une activité.

La deuxième partie du modèle RM-ODP [2] donne la définition suivante:

"activité: graphe d'actions sans cycle et avec une seule racine, dans lequel l'occurrence de toute action du graphe est conditionnée par l'occurrence de toutes les actions qui la précèdent immédiatement (c'est-à-dire toutes les actions adjacentes qui sont plus proches de la racine)."

6.2.7 Services

Un service est le résultat de la négociation d'un ensemble donné de caractéristiques de service entre le fournisseur et l'utilisateur du service, en fonction de l'expression des besoins de l'utilisateur et des capacités du fournisseur. Un service n'est PAS la gestion du service au sens du RGT. Un service est un ensemble d'opérations permettant d'atteindre un objectif donné et peut exister à tout niveau de l'architecture logique des couches.

6.2.8 Caractéristique de service

Une caractéristique de service est un ensemble de règles de gestion communes, d'activités et/ou d'actions et des règles de gestion associées.

6.2.9 Combinaison des services

La définition de contrats basés sur d'autres contrats appelle une étude ultérieure.

6.3 Extension de communautés

La méthode d'extension de communautés appelle une étude ultérieure.

6.4 Domaines

Le concept de domaine RM-ODP, tel qu'il est défini au 10.3 de la deuxième partie du modèle RM-ODP [2] est lié à l'établissement de communautés. Il est prévu que les domaines fassent partie de la méthodologie; toutefois les concepts nécessaires doivent être affinés en fonction du modèle de niveau réseau. Il s'ensuit que l'application de ce concept dans la présente méthodologie appelle une étude ultérieure.

7 Perspective informationnelle

La perspective informationnelle est définie dans la troisième partie du modèle RM-ODP [3]. Elle définit les types d'information au sein d'un système réparti. La perspective informationnelle indique, du point de vue de la tâche de modélisation du niveau réseau, les aspects concernant les informations liées aux ressources gérées et aux application de gestion (y compris les états et les transitions significatives).

La perspective informationnelle définit des types d'objets d'information, les relations entre ces types, leurs attributs et leurs états, y compris les transitions autorisées entre états. La perspective informationnelle définit, du point de vue de la modélisation au niveau réseau, des ensembles statiques et invariants d'information (schéma) liés aux ressources gérées (par exemple les connexions, les liaisons, les points de terminaison du réseau) représentées par le modèle. Elle définit également le schéma dynamique des ressources décrivant les changements d'état autorisés pour un ou plusieurs objets d'information.

La spécification de la perspective informationnelle se constitue de trois parties:

- une description non formalisée spécifiée en langage naturel et utilisant des étiquettes adéquates de mots clés (par exemple DEFINITIONS, INVARIANTS, etc.);
- une description semi-formalisée qui est spécifiée en utilisant un sous-ensemble des directives de définition des objets gérés (GDMO) [5] et le modèle général de relations (GRM) [6] tel qu'il est utilisé par la gestion OSI;
- des définitions faites en utilisant la notation Z [7].

La liste des relations potentielles est donnée par surcroît dans la perspective informationnelle commune afin d'en améliorer la lisibilité.

L'Annexe B donne une description complète des squelettes de la perspective informationnelle.

La méthode d'expression dans la perspective informationnelle n'impose de contrainte ni à l'accès fourni dans la perspective informatique ni à la représentation donnée dans la perspective d'ingénierie.

7.1 Perspective informationnelle

La Recommandation G.853.1 [10], Eléments communs de la perspective informationnelle pour la gestion d'un réseau de transport, ou perspective informationnelle commune, contient la définition des objets d'information et des relations qui représentent les ressources au sens de G.805, d'une manière

indépendante de tout service de gestion particulier. Des attributs d'information communs et des états communs sont également spécifiés.

La perspective informationnelle commune fournit la base pour l'élaboration de perspectives informationnelles spécifiques de chaque application de gestion.

Lorsque l'expression des besoins est faite pour une application de gestion donnée (par exemple la gestion de la connexion), ceux-ci sont définis pour une communauté entreprise et les perspectives informationnelles spécifiques de l'application de gestion sont ensuite élaborées. La Recommandation G.853.1 fournit la base à partir de laquelle une telle perspective spécifique de l'application de gestion peut être élaborée.

Les objets d'information spécifiques d'une application de gestion peuvent être créés au moyen de sous-classes à partir des objets de la perspective informationnelle commune faisant l'objet d'une extension pour les besoins de cette application. Dans ce cas, les sous-classes spécifiques de la nouvelle application de gestion peuvent contenir d'autres attributs de la perspective informationnelle commune en plus de ceux définis dans sa superclasse. D'autres relations et d'autres attributs peuvent également être créés pour cette application en cas de besoin. Il est également possible d'ajouter de nouveaux objets hérités de la classe sommitale de l'information réseau networkInformationTop.

Si les définitions d'attributs sont compatibles avec les attributs d'objets gérés GDMO existants (par exemple dans la Recommandation G.774 [11]), la référence à ces attributs sera fournie d'une manière non formelle. Dans ce cas, la spécification de la perspective informationnelle importe la sémantique de l'attribut, mais non sa syntaxe (qui peut être importée à partir de la perspective informatique correspondante).

Des squelettes modifiés de modèle général de relation (GRM) font partie de la présente Recommandation afin d'indiquer de quelle manière les objets sont en relation. Chaque squelette GRM identifie des rôles de la relation ainsi que les objets susceptibles de jouer ces rôles. Les relations, définies initialement dans la spécification de la perspective informationnelle commune et auxquelles peut participer un objet d'information, sont énumérées dans la partie de l'objet décrivant ses relations potentielles. Lorsqu'un objet de perspective informationnelle commune fait l'objet d'une spécialisation en sous-classes pour une perspective informationnelle spécifique de l'application de gestion, les relations qui sont considérées comme nécessaires pour cette application sont déclarées obligatoires.

La Recommandation G.853.1 [10] contient également des attributs communs pouvant être inclus lorsque les sous-classes spécifiques de l'application de gestion sont créées. Des exemples de tels attributs sont `operationalState` (*état opérationnel*) et `userLabel` (*étiquette utilisateur*).

8 Perspective informatique

La perspective informatique est définie dans la troisième partie du modèle RM-ODP [3]. Elle définit la décomposition fonctionnelle du système en objets qui interagissent à travers des interfaces spécifiées de manière à faciliter la répartition. Dans la perspective informatique, les applications sont représentées par des configurations d'objets qui interagissent. Ces interfaces, définies par la perspective informatique, indiquent le niveau maximal de répartition des objets pouvant être pris en charge. La décision finale concernant le niveau effectif de répartition pris en charge dans un système ouvert est définie dans la perspective d'ingénierie. La spécification informatique des interactions entre objets est définie en outre, pour chaque objet, par les interfaces détaillées qu'il fournit ainsi que par sa signature opérationnelle et sa spécification de comportement. Ces spécifications font référence aux transitions d'état des objets d'information en utilisant les correspondances de paramètres.

Une conception complète de la perspective informatique impose de définir les objets informatiques, en plus de la définition des interfaces et des opérations. Contrairement à une définition limitée aux interfaces, la définition des objets permet de spécifier l'interaction entre interfaces aussi bien que le nombre et les types (client/serveur) des interfaces devant être prises en charge par un objet donné. Ceci fournit une définition complète de l'application en relation avec l'objet. La définition d'un objet donné possédant plusieurs interfaces évite d'avoir à traiter des objets multiples dupliquant l'état d'une ressource spécifique ou d'une autre entité gérée. Si une interface d'objet doit être répartie, cet objet peut être décomposé en plusieurs objets informatiques.

La perspective informatique spécifie des objets informatiques, des interfaces et des opérations définies comme suit:

- les **objets informatiques** sont définis comme une vue particulière de l'information définie dans les objets d'information aux fins spécifiques d'une application. Un objet informatique spécifie les interfaces serveur et client associées à l'objet et le comportement de cet objet (c'est-à-dire les contraintes entre interfaces). La clause de comportement décrit les relations entre les interfaces de l'objet. Elle peut également décrire le rôle de cet objet lors de la fourniture du service au niveau de chacune de ses interfaces ainsi que tout état initial spécifique interne à l'objet;
- les **interfaces** définissent un ensemble d'opérations pouvant être invoquées au niveau de l'interface ainsi que le comportement de l'interface. La clause de comportement décrit le service fourni par l'interface. Elle peut également spécifier des contraintes d'ordre (ou de succession) pour les opérations définies par l'interface;
- les **opérations** sont définies par leurs signatures. Une signature d'opération contient des paramètres d'entrée et de sortie, des préconditions et des postconditions, des exceptions pouvant être activées et le comportement de l'opération définissant la sémantique de l'opération. Les opérations peuvent faire référence à des données qui ont été définies dans des objets d'information.

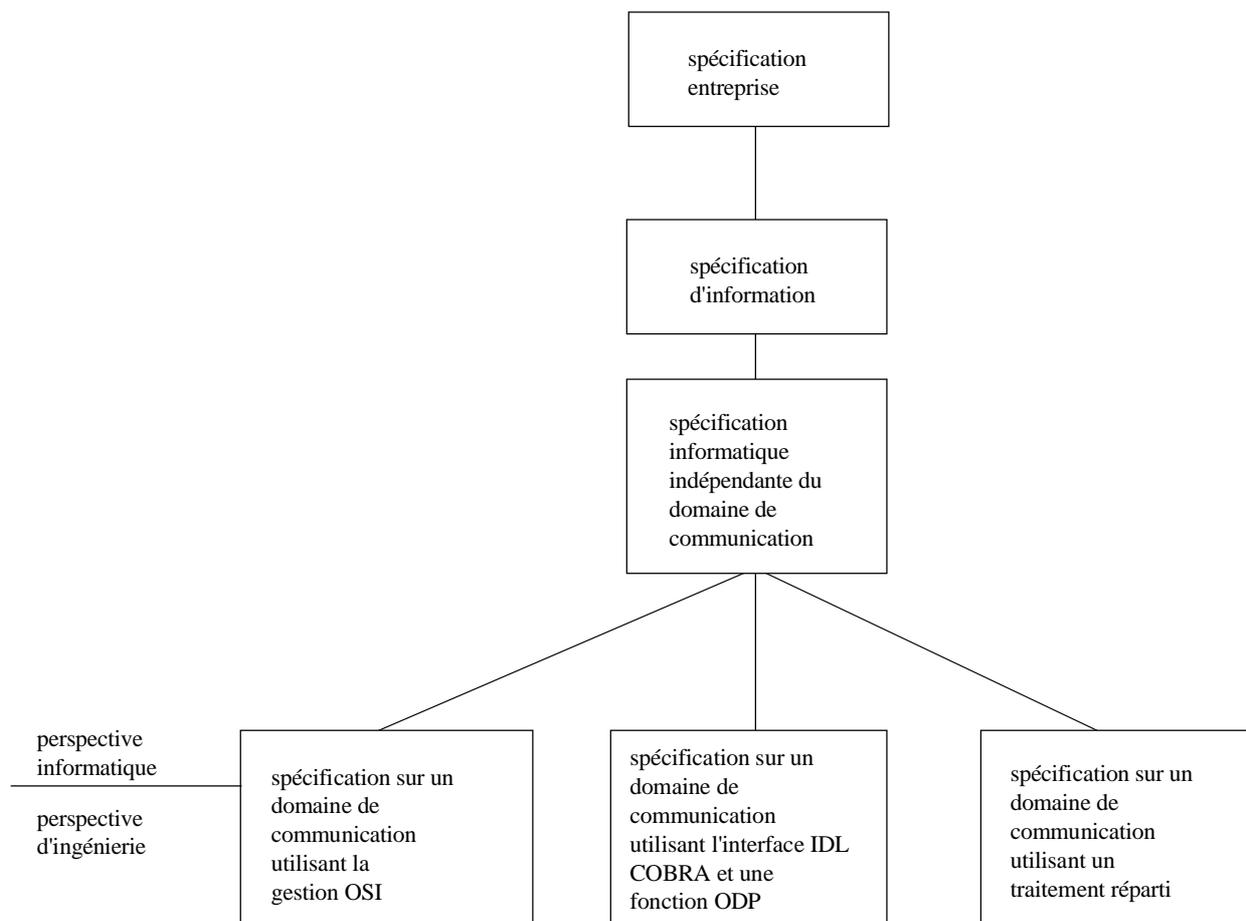
8.1 Concepts informatiques

- Les objets informatiques interagissent à travers des interfaces.
- Un objet informatique donné peut posséder de multiples interfaces et des instances multiples du même type d'interface.
- Les changements d'état d'un système résultant d'opérations au niveau d'une interface peuvent être vus sur les autres interfaces du même objet ou sur des interfaces vers d'autres objets.
- Les objets informatiques ne sont définis pour une application que si les interactions requises entre les interfaces ont besoin d'être normalisées.
- Une interface informatique peut disposer d'opérations qui fournissent des fonctions d'observation et de manipulation de l'information concernant les ressources gérées. Cette information peut avoir été définie dans la perspective informationnelle et fait l'objet de références au moyen des paramètres de l'opération. Les types des données impliquées sont les objets d'information définis dans la perspective informationnelle. Ces données sont regroupées sur décision des applications particulières, l'instanciation et l'accès sont fournis par des interfaces d'opération décrites dans la perspective informatique.

8.1.1 Perspective informatique indépendante du domaine de communication

Comme indiqué par la Figure 2, la perspective informatique peut être divisée en deux parties du point de vue de la présente méthodologie: une partie qui est indépendante de l'environnement d'ingénierie ou du domaine de communication sous-jacent et une partie qui est étroitement liée au domaine de communication utilisé dans la perspective d'ingénierie. La méthode de spécification de la

perspective informatique indépendante de la communication est définie dans l'Annexe C. La présente Recommandation est, autant que possible, indépendante du domaine de communication choisi pour la réalisation de la perspective d'ingénierie sous-jacente (par exemple CMIP/OSI ou CORBA).



T1521050-96

Figure 2/G.851.1 – Variantes de spécifications de domaines de communication

8.1.2 Mappages vers la spécification de la perspective informatique dépendante du domaine de communication

Les squelettes définis dans l'Annexe C doivent être mappés vers les squelettes du domaine de communication utilisés dans la réalisation d'ingénierie. La définition de la syntaxe des paramètres d'opération informatique utilise un sous-ensemble de la syntaxe abstraite ASN.1. Cette représentation par une syntaxe abstraite, ainsi que les opérations informatiques, doivent être mappées vers la syntaxe et le protocole propres au domaine de communication utilisé dans la réalisation d'ingénierie. Ce processus fournit comme résultat une spécification informatique qui est spécifique de chaque domaine de communication envisagé.

9 Perspective d'ingénierie

9.1 Introduction

Une spécification d'ingénierie définit, dans le modèle RM-ODP, l'infrastructure devant prendre en charge la répartition fonctionnelle d'un système ODP. Il est fait l'hypothèse qu'une transparence vis-à-vis de la répartition est obtenue par l'assemblage de représentations ODP transparentes de fonctions et d'objets. Cette méthodologie spécifie au départ un processus d'ingénierie permettant de réaliser l'implémentation de la répartition fonctionnelle pour les deux domaines de communication constitués par la gestion OSI et par l'utilisation du langage IDL CORBA avec des fonctions ODP. D'autres domaines de communication (infrastructures) peuvent être pris en considération ultérieurement.

La Figure 2 illustre ces possibilités.

9.2 Concepts d'ingénierie

La perspective d'ingénierie du modèle RM-ODP fournit les mécanismes de prise en charge de la transparence de la perspective informatique. Alors que la spécification informatique est indépendante du protocole et de l'infrastructure, la perspective d'ingénierie est spécifique de l'infrastructure et satisfait à des scénarios dans lesquels un choix a été fait pour les interfaces et dans lesquels les exigences de transparence sont respectées.

Si au cours de la définition de la perspective d'ingénierie, on découvre des prescriptions supplémentaires qui influent sur les autres perspectives, il faut mettre à jour les perspectives entreprise, informationnelle et informatique en conséquence.

La perspective entreprise doit énoncer quels sont les scénarios à prendre en charge. Un scénario indique avant tout les besoins entreprise à satisfaire et les considérations qui conduisent aux choix des interfaces. La décision de regrouper des objets informatiques peut résulter de contraintes d'environnement telles que des exigences de temps réel (par exemple un délai maximal pour une interaction entre objets).

La technique du modèle Ensemble décrite au paragraphe 10 peut être utilisée comme base pour la spécification d'objets d'ingénierie qui satisfont aux besoins entreprise et aux contraintes d'environnement d'un scénario donné. La technique du modèle Ensemble convient pour la définition d'un scénario satisfaisant aux besoins entreprise.

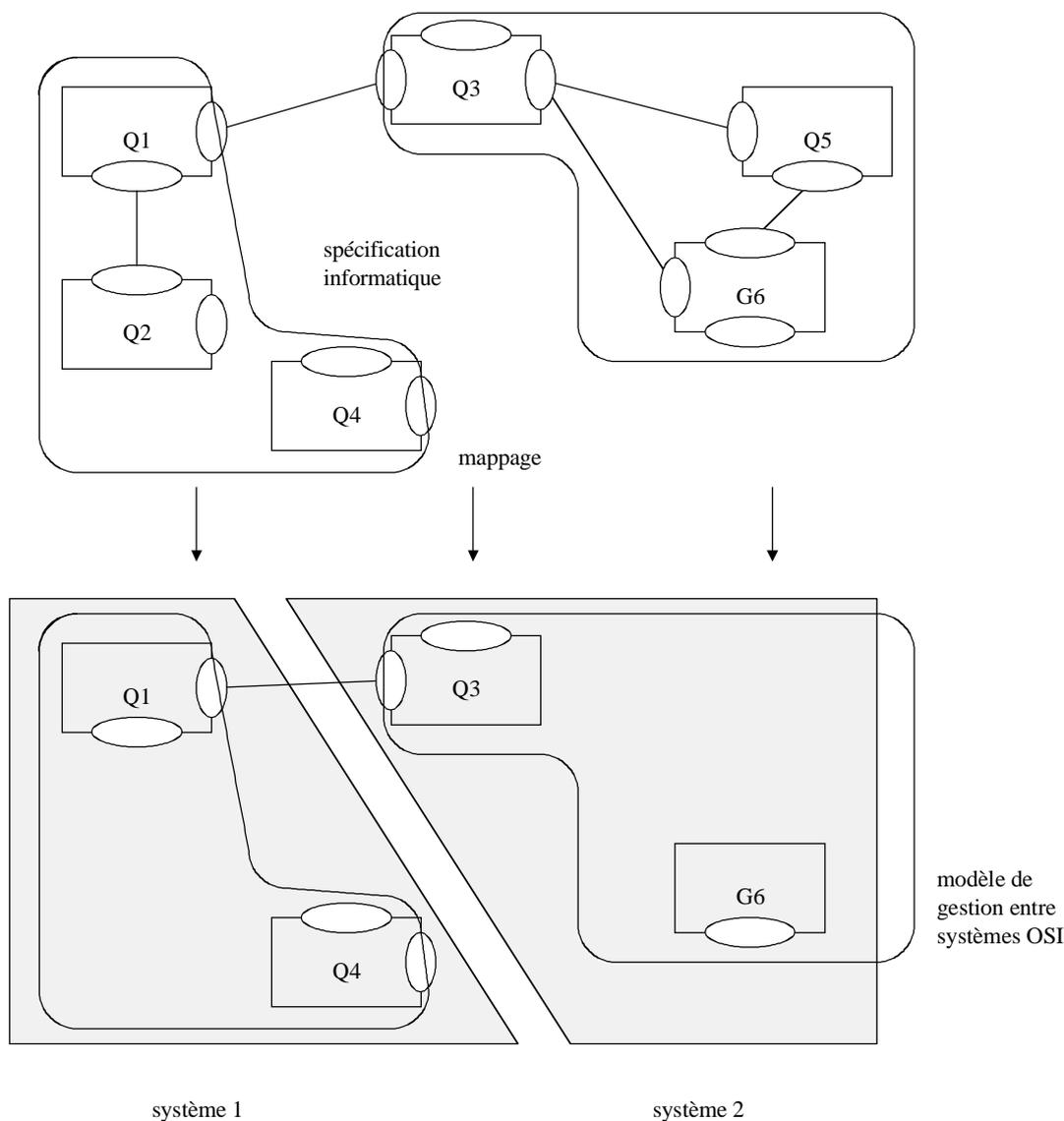
Les versions d'exécution (objets d'ingénierie de base – BEO) des objets informatiques peuvent être assemblées en agrégats qui satisfont aux exigences du modèle Ensemble. Un agrégat peut être considéré comme un bloc de construction de base constituant une entité unique en ce qui concerne la migration, les reprises, l'initialisation ou la duplication. Un agrégat ODP s'exécute sur un seul système physique. Un agrégat offre à la fois des interfaces client et serveur qui sont prises en charge par des souches et des liens spécifiques du protocole. Un mappage existe même si un objet BEO n'est pas nécessairement équivalent à un objet géré.

Squelettes de mappage de relation: la définition complète d'une relation n'existe qu'une fois que les squelettes de mappage des relations GRM sont définis. Ces liaisons font partie de la perspective entreprise, car elles sont spécifiques de l'infrastructure.

9.3 Perspective d'ingénierie basée sur la gestion OSI

Ce sous-paragraphe ne considère que les agrégats et les implémentations basés sur la gestion OSI. Le modèle d'information de gestion OSI définit une relation entre un gestionnaire et un agent situés dans deux systèmes OSI distincts. Les spécifications informatiques ne font pas d'hypothèses au sujet de la

répartition physique des objets, mais cette spécification d'ingénierie doit toutefois être considérée dans le contexte du RGT. Ceci implique que l'ensemble d'objets informatiques défini doit être scindé en deux parties, chacune d'elles étant localisé sur un système OSI qui lui est propre. Dans ce cas, les objets informatiques peuvent être assemblés en deux groupes qui n'offrent que ceux des interfaces informatiques qui doivent être vus sous la forme d'interfaces de gestion-système OSI, comme indiqué par la Figure 3 par l'agrégat associé.



T1521060-96

Figure 3/G.851.1 – Regroupement d'objets informatiques permettant de définir des interfaces de systèmes de gestion OSI

9.3.1 Influence des protocoles sur les perspectives informationnelle et informatique

Les modèles informatique et d'ingénierie doivent être aussi indépendants l'un de l'autre que possible. Le niveau informatique sera toutefois influencé par des contraintes d'ingénierie (par exemple l'utilisation de l'information SMI). En conséquence, la perspective informationnelle devra également prendre en charge l'utilisation de l'information SMI (par exemple en définissant des relations GRM avec mappage de pointeurs dans les objets gérés). Ceci permettra un mappage des opérations informatiques vers des actions ou des attributs d'élément CMISE. Cette extension de l'information

fait partie du processus d'ingénierie et conduit, au niveau des interfaces d'ingénierie, à l'élaboration d'un modèle d'information tenant compte des points suivants:

- la spécification d'interfaces informatiques indépendantes;
- les exigences de répartition;
- les capacités de communication (par exemple l'élément CMISE).

9.3.2 Contraintes d'ingénierie imposées par la perspective entreprise

La répartition de services de gestion sur plusieurs systèmes peut être formulée comme une préoccupation entreprise qui impose des contraintes d'ingénierie (par exemple la définition d'une interface de gestion vers un élément réseau normalisé constituant une abstraction de ressources réparties). Il est possible de décrire différents scénarios pour une spécification entreprise des contraintes d'ingénierie, conduisant à des profils et des modèles Ensemble différents.

9.3.3 Utilisation de paquetages et d'objets gérés GDMO

Des prescriptions sont nécessaires pour faire un choix entre différentes manières d'exprimer l'information du niveau ingénierie au moyen des directives GDMO.

Les ressources (transport et ressources gérées) peuvent être modélisées sous la forme d'objets gérés, car elles représentent des entités pouvant être manipulées par tout service de gestion.

Les services de gestion peuvent être modélisés de l'une des deux manières suivantes:

- 1) un service de gestion donné peut être modélisé sous la forme d'un objet géré et la combinaison avec la ressource gérée peut être faite par héritage au moyen d'un objet géré dérivé de la classe d'objets gérés et de la classe d'objets de service de gestion. Cette solution conduit à un grand nombre de classes d'objets gérés. Elle diffère en outre de la combinaison des interfaces définie dans le modèle RM-ODP (en particulier pour ce qui est des liaisons dynamiques). L'utilisation de pointeurs combinés entre la ressource gérée et le service de gestion est en accord avec la notion de combinaison des interfaces, mais cette solution conduit toutefois à un nombre élevé de pointeurs;
- 2) un service de gestion donné peut être modélisé sous la forme d'un paquetage et la combinaison avec les ressources gérées peut être faite par inclusion conditionnelle du paquetage dans la classe d'objets gérés représentant la ressource. Cette solution est en accord avec l'utilisation de bibliothèques d'ingénierie prédéfinies. Cette bibliothèque contiendra des objets gérés représentant des ressources de transport (c'est-à-dire correspondant à la perspective informationnelle commune des objets d'informations) et, pour chacune d'elles, d'un paquetage par service potentiel pris en charge. Ceci évitera les réécritures et fournira une vue plus homogène du système.

Il convient de noter que de multiples instances d'un même type peuvent être définies dans la perspective informatique. L'utilisation de paquetages GDMO ne convient pas dans ce cas.

9.3.4 Prise en charge de services de gestion multiples au niveau de l'interface d'ingénierie

Les paquetages peuvent être totalement indépendants entre eux et peuvent dans ce cas être inclus dans l'objet géré lors de la spécification du service correspondant pour l'interface d'ingénierie. Certains paquetages peuvent toutefois être sujets à des interdépendances. La meilleure manière de procéder dans ce cas consiste à définir un paquetage composé, mais il est alors nécessaire de définir le service composé correspondant et d'en énoncer les interdépendances dans les perspectives entreprise, informationnelle et informatique. Certains paquetages peuvent en outre s'exclure mutuellement au sein d'un objet géré. Ceci doit être exprimé dans l'énoncé du comportement de l'objet géré.

9.3.5 Identification

Les interfaces informatiques doivent être identifiées conformément au modèle RM-ODP afin de pouvoir faire l'objet de références (par exemple pour l'établissement des liaisons). Les interfaces informatiques peuvent être mappés vers des paquetages. Les paquetages souhaités seront instanciés en même temps que les objets gérés qui les prennent en charge. Une référence d'ingénierie unique sera créée pour l'objet géré. Cette référence d'ingénierie peut être indépendante de toute préoccupation d'information et créée, en conséquence, selon des règles de gestion qui ne sont connues que par le système d'ingénierie qui les prend en charge. Une autre solution est de définir un attribut instanceId d'identification de l'instance qui véhicule cette information.

9.3.6 Mappage de relation

Les relations GRM seront mappées vers les squelettes ROLE BINDING dans la perspective entreprise. La représentation peut être faite, par exemple, sous la forme de pointeurs, de corrélations de nom, d'opérations ou d'héritage.

9.3.7 Comparaison entre l'utilisation d'actions par rapport à des opérations d'attribut pour des modifications de relations et d'état

L'élément CMISE permet de modifier l'état d'un système de deux façons:

- opérations REPLACE sur des attributs;
- actions.

Les actions seront utilisées lorsque des attributs multiples sont impliqués et que l'opération est considérée comme élémentaire, ou lorsque son comportement significatif est exigé. Une opération REPLACE peut être prescrite si un seul attribut est modifié et qu'il n'est pas nécessaire de fournir un accès en écriture à l'attribut.

9.4 L'environnement de traitement réparti

9.4.1 Aperçu général

L'utilisation d'un environnement de traitement réparti (DPE) appelle une étude ultérieure. Il est possible de donner les considérations suivantes:

- comme il existe de multiples choix d'implémentations (par exemple OSI, CORBA, environnements DPE basés sur l'environnement DCE), l'ensemble des modèles d'ingénierie doit utiliser un environnement DPE commun autorisant une interaction entre ces domaines;
- des formes locales de dénomination, de protocole et d'architecture de fiabilité peuvent être utilisées au sein de chaque domaine, mais un ensemble de services support est requis pour l'interfonctionnement entre domaines. Par exemple, un seul négociant sera approuvé par la Commission d'études 15;
- les objets d'ingénierie de base peuvent être pris en charge par des objets de souche et de protocole afin de faciliter l'interfonctionnement entre objets sélectionnés par divers protocoles.

Il existe également un besoin de définir des prescriptions de transparence pour l'environnement DPE utilisé par les applications de gestion du réseau de transport.

10 Utilisation d'ensembles dans la définition d'applications de gestion de réseau

10.1 Domaine d'application

La méthodologie de la présente Recommandation fournit des spécifications propres à chaque application pour les perspectives entreprise, informationnelle et informatique qui sont en grande partie indépendantes de l'infrastructure de répartition. La technique du modèle Ensemble est utilisée pour permettre la définition d'une solution d'ingénierie qui satisfasse au scénario d'application. Ceci est illustré par la Figure 4.

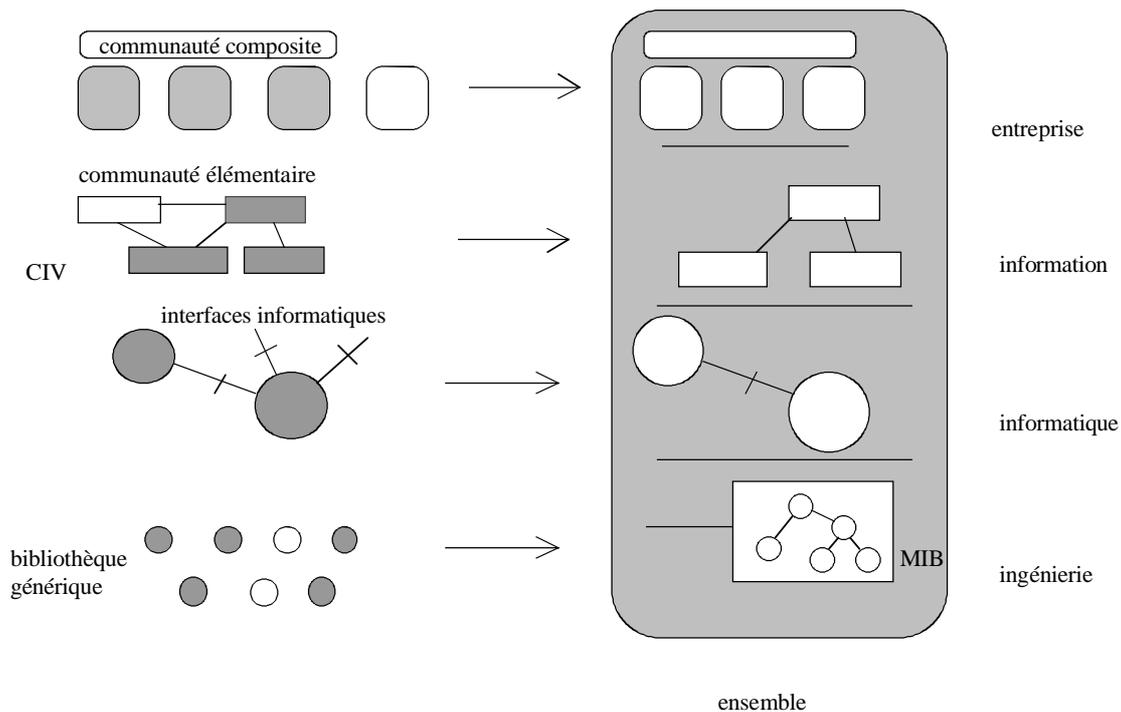


Figure 4/G.851.1 – Utilisation d'un modèle Ensemble pour l'élaboration d'une interface spécifique d'application

Le modèle Ensemble identifie un ensemble d'applications devant être prises en charge (du point de vue des communautés entreprise), ainsi que d'autres contraintes d'implémentation telles que le niveau de distribution à fournir, le niveau de visibilité et le protocole d'interface.

Le modèle Ensemble ne fait partie d'aucune perspective du modèle RM-ODP mais fait référence aux perspectives entreprise, informationnelle, informatique et d'ingénierie afin de permettre l'élaboration d'une solution satisfaisant aux besoins d'un ensemble de communautés d'entreprise. Le modèle Ensemble est un guide d'utilisation du contenu des perspectives du modèle RM-ODP en vue de résoudre un problème de gestion particulier. Ceci signifie que le modèle Ensemble fournit une solution particulière en fonction d'une spécification d'interface d'ingénierie qui impose une répartition donnée de la fonctionnalité de gestion. La conformité sera testée au niveau de la perspective entreprise.

Le squelette du modèle Ensemble est défini dans l'Annexe F.

ANNEXE A

Squelette et directives générales pour la spécification de la perspective entreprise

La présente annexe donne une explication concernant la spécification de la perspective entreprise.

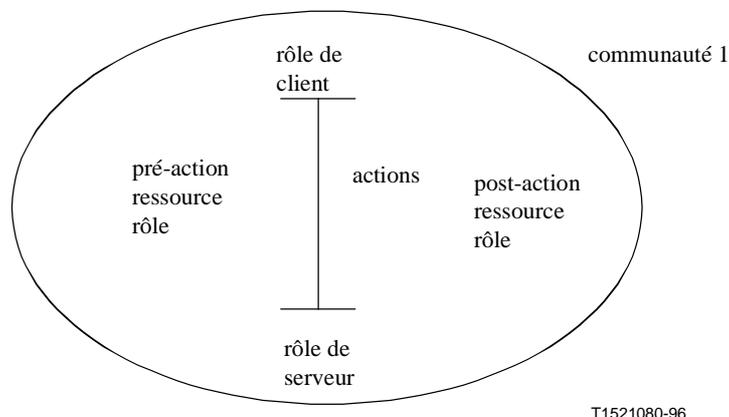


Figure A.1/G.851.1 – Les membres d'une communauté ont des rôles par rapport à des actions

A.1 Définition non formelle du squelette entreprise

X COMMUNAUTE <community_label> "Nom"

X.1 BUT

community_definition -- Cette clause présente le but de la communauté.

X.2 ROLE

<community label><Role Label>

role_definition -- Fournit la liste des rôles des membres impliqués dans la communauté ainsi qu'une brève description.

X.3 REGLES DE GESTION

PERMISSION <label_string>|OBLIGATION <label_string>|PROHIBITION
<label_string>|EXCEPTION <label_string>

policy_definition -- Fournit les règles de gestion s'appliquant à l'ensemble de la communauté.

X.4 ACTION

X.4.1 <action label> "Nom de l'action"

action_definition -- Fournit la liste des actions prenant en charge le but de la communauté.

ACTION POLICY: -- Fournit une description brève, claire et concise des règles de gestion sous l'une des formes OBLIGATION, PERMISSION, PROHIBITION ou EXCEPTION.

PERMISSION <label_string>|OBLIGATION <label_string>|PROHIBITION
<label_string>|EXCEPTION <label_string>

policy_definition -- *Chacune de ces règles stipule le rôle et l'information impliquée. La perspective entreprise n'a pas l'intention de fournir de prescriptions au sujet de cette information.*

X.5 ACTIVITE

Chacune des activités est définie dans une subdivision portant comme titre le nom de l'activité et une introduction présentant la définition de l'activité.

Nom de l'activité <activity label>

activity_definition

WITH ACTION GRAPH -- *Liste des actions élémentaires.*

Start

<action label>

<nth action label>

End

ACTIVITY POLICY -- *Ce graphe d'actions doit garantir que les interdépendances, telles que l'ordre de succession des actions, sont stipulées de manière claire en tant que règle de gestion.*

PERMISSION <label_string>|OBLIGATION <label_string>|PROHIBITION
<label_string>|EXCEPTION <label_string>

policy_definition

X.6 CONTRAT

Un service est une négociation d'un ensemble de caractéristiques faite entre le fournisseur du service et l'utilisateur du service en fonction des besoins de l'utilisateur et des possibilités du fournisseur. Le résultat de la négociation est un contrat indiquant un accord au sujet de la sélection d'un ensemble de caractéristiques offertes par le fournisseur. Cette phase d'établissement du contrat appelle une étude ultérieure, toutefois les caractéristiques du service peuvent être fixées sous la forme d'un ensemble d'actions prises en charge et, lorsque cela s'applique, de règles de gestion prises en charge.

A.2 Définition formelle du squelette entreprise

Le présent sous-paragraphe fournit une description formelle de la perspective entreprise en notation BNF.

NOTE – Cette spécification étend la notation BNF comme suit:

- [...] élément optionnel, zéro ou une occurrence
- * élément d'une liste, une ou plusieurs occurrences
- *le reste de la ligne est un commentaire comme dans la notation ASN.1*

```
<community_template> ::=  
"COMMUNITY" <label> <name>
```

```
<doc_heading> "PURPOSE" <community_definition>
```

```
<doc_heading> "ROLE"  
<role_definition>*
```

```

<doc_heading> "POLICY"
  <policy_definition>*

<doc_heading> "ACTION"
  <action_description>*

<doc_heading> "ACTIVITY"
  <activity_definitions>*

["WITH ACTION GRAPH"
  "Start"
  <action_label>*
  "End"]

"CONTRACTS" <label_string>

<action_description> ::= <doc_heading> <label> <label_string>
  "ACTION POLICY" ":"
  <policy_definitions>

<action_label> ::= <label>

<activity_definition> ::= <doc_heading> <label>
  <label_string>
  "ACTIVITY POLICY"
  <policy_definitions>

<activity_definitions> ::= "None"
  | <activity_definition>*

<community_definition> ::= <label_string>

<doc_heading> ::= <text>
  -- doc_heading est une en-tête utilisée lors de l'impression du squelette
  -- Cette information est spécifique du système de publication
  -- et considérée comme des caractères blancs dans une implémentation

<label> ::= "[A-Za-z][-A-Za-z0-9.]*"

<label_string> ::= -- quoted string | label

<name> ::= <label>

<policy_definitions> ::= "None"
  | <policy_definition>*

<policy_definition> ::= { "PERMISSION" | "OBLIGATION" | "PROHIBITION" |
  "EXCEPTION" } <label> <label_string>

<role_definition> ::= <role_label> <label_string>

<role_label> ::= <label>

```

Structure de la perspective informationnelle

B.1 Introduction

La section perspective informationnelle est constituée des sous-sections suivantes:

- une liste des entités d'information qui ont été définies dans d'autres spécifications et sont référencées par la communauté entreprise correspondante, incluant, à des fins d'héritage, d'autres entités d'information en provenance d'autres spécifications. Lorsque des entités d'information sont importées de cette manière dans une spécification, toutes les autres spécifications se rapportant à ces entités sont également importées (c'est-à-dire que les spécifications non formelles, semi-formelles et formelles sont importées lorsque l'une d'elles est mentionnée);
- Des diagrammes d'objet d'information et de types de relation, lorsqu'ils sont nécessaires pour améliorer la lisibilité:
 - définissent le contenu des diagrammes rôle² relation et classe rôle. La distinction entre rôle et classe est basée sur l'utilisation des notations des directives GDMO et GRM dans lesquelles des relations sont définies entre les rôles pouvant être joués par des instances de classes d'objets;
 - des diagrammes d'héritage;
 - des diagrammes de relation;
 - des diagrammes entité-relation.
- Des classes d'objets d'information:
 - définissent les classes d'objets. Les descriptions semi-formelles utilisent un squelette INFORMATION OBJECT CLASS (*classe d'objets d'information*) dont la structure et la syntaxe sont similaires à celles du squelette GDMO MANAGED OBJECT CLASS (*classe d'objets gérés*) défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4 (GDMO) [5]. Les définitions formelles utilisent la notation Z [7].
- Des relations entre informations:
 - définissent les relations auxquelles peuvent participer les objets d'information. Utilisent le squelette RELATIONSHIP CLASS (*classe de relations*) défini selon la Rec. UIT-T X.725 | ISO/CEI 10165-7 (GRM) [6] pour la partie semi-formalisée et la notation Z pour la partie formelle.
- Des définitions de schéma statique:
 - définissent les états globaux et statiques du système décrits dans les schémas. Cette sous-section contient des parties non formalisées, semi-formalisées et formelles.
- Des définitions de transition de schéma (schémas dynamiques):
 - attribuent des étiquettes à des transitions particulières entre états composés, décrites par un schéma statique, de sorte que ces transitions peuvent faire l'objet de références comme déclencheurs de notifications. Cette sous-section contient des parties non formalisées, semi-formalisées et formelles.

² Au sens de la directive GRM.

- Des définitions de type d'attribut:
 - définissent les attributs pour le service concerné. Utilisent le squelette ATTRIBUTE défini selon la directive GDMO pour la partie semi-formalisée et la notation Z pour la partie formelle.

B.2 Descriptions du modèle

Trois spécifications sont fournies pour toute définition contenue dans la section perspective informationnelle:

- Une description non formalisée (en langage naturel) qui est prévue pour pouvoir être lue par un humain. Afin d'obtenir la plus grande concision possible, la spécification est structurée en utilisant des étiquettes pour présenter chaque paragraphe (par exemple: <definition>, <invariants>, ...) et ses sous-paragraphe (<inv_1>, <operationalState>, ...). Ces étiquettes sont utilisées pour fournir des références croisées entre diverses perspectives d'un service donné ou des références croisées entre plusieurs services.
- Une description semi-formalisée utilisant une notation dérivée de la directive GDMO pour les classes d'objets et les attributs. Cette notation GDIO (Directive générale pour la définition des objets d'information) est basée sur l'utilisation des structures de classe d'objets gérés et de squelette d'attribut de la directive GDMO. Dans cette spécification formelle, tout langage naturel (utilisé normalement dans le paragraphe BEHAVIOUR) sera remplacé par le contenu équivalent de la description non formalisée. Cette description utilise la notation de la directive GRM pour les squelettes de relation. Le paragraphe BEHAVIOUR fera également référence à la partie non formalisée.
- Une description formelle utilisant la notation Z.

B.3 Structure de la spécification

B.3.1 Classes d'objets d'information

B.3.1.1 Description non formalisée

La partie non formalisée de la spécification de la classe d'objet est structurée en quatre parties:

- la première partie, débutant par le mot clé <definition>, contient une définition de la classe d'objets d'information;
- la deuxième partie, débutant par le mot clé <attributes>, donne la liste des attributs de la classe d'objets d'information par inclusion d'une référence, pouvant être locale si l'attribut est défini localement, ou globale si l'attribut est importé à partir d'un autre service;
- la troisième partie, débutant par le mot clé <invariants>, décrit les états autorisés pour une instance de la classe d'objets d'information. Ceux-ci peuvent être exprimés sous la forme de valeurs d'attribut individuelles, ou comme combinaisons d'attributs si ces derniers sont interdépendants. Les interdépendances de valeurs d'attribut peuvent également être décrites dans un schéma statique (voir B.3.3). Les valeurs autorisées peuvent être décrites d'une manière indépendante si l'état d'un attribut ne dépend pas des états des autres attributs. Toutes les valeurs sont autorisées par défaut si la spécification des valeurs autorisées est absente;
- la quatrième partie, débutant par le mot clé <transitions> exprime les transitions d'état entre l'ensemble des états combinés de l'objet d'information. La spécification des transitions d'état requise dans cette partie peut être influencée par la spécification d'invariants faite dans la partie <invariants>. Un invariant peut, par exemple, limiter la visibilité du besoin de certaines autres transitions d'état possibles.

Les transitions d'un attribut peuvent être décrites d'une manière indépendante si un attribut est indépendant des autres, c'est-à-dire si ses valeurs n'influent pas sur les transitions d'état des autres attributs. En l'absence de cette définition, toutes les transitions d'état sont autorisées par défaut pour cet attribut.

B.3.1.2 Description semi-formalisée

Cette description utilise la notation GDMO en remplaçant le mot clé "MANAGED OBJECT CLASS" par "INFORMATION OBJECT CLASS" et avec les restrictions suivantes:

- les attributs ne peuvent pas avoir de spécification d'accès (*access-spezifier*);
- la clause REGISTERED AS n'est pas exigée;
- aucune clause ACTIONS ou NOTIFICATIONS ne peut être spécifiée;
- la clause WITH ATTRIBUTE SYNTAX ne sera pas utilisée dans le squelette ATTRIBUTE;
- l'expression GDMO "BEHAVIOUR" sera structurée comme suit, conformément aux sous-paragraphes contenues dans la description non formelle:
 - la partie définition est référencée par inclusion du mot clé <definition>;
 - les parties <invariants> et <transitions> sont copiées dans la clause BEHAVIOUR;
 - les sous-classes spécifiques d'une application indiqueront les relations qui sont effectivement utilisées.

La partie <attributes> de la description non formalisée doit être convertie en une expression ATTRIBUTES du squelette d'objet d'information ne contenant aucune spécification d'accès.

B.3.1.3 Description formelle

La description formelle utilise la notation Z. L'Appendice VII donne un exemple d'application de la notation Z pour cette description.

B.3.1.4 Relations potentielles

Cette partie de la définition de l'objet d'information n'est exigée que pour la spécification de la perspective informationnelle commune. Elle définit les relations possibles auxquelles peut participer l'objet d'information ou ses sous-classes. Elle ne fait partie de la perspective informationnelle commune que dans un but de clarté de la spécification et ne constitue pas une liste des relations qui doivent être prises en charge par l'objet d'information ou ses sous-classes et n'est pas nécessairement exhaustive.

Lorsque l'objet perspective informationnelle commune est détaillé en sous-classes pour les besoins d'une perspective informationnelle spécifique d'une application de gestion, les relations nécessaires pour cette application, et qui affectent le comportement de l'objet, sont énumérées dans la clause BEHAVIOUR de la description semi-formalisée de la nouvelle sous-classe ainsi que dans la partie relation de la description non formalisée. Ces relations peuvent être un sous-ensemble de celles qui sont énumérées dans la superclasse de la perspective informationnelle commune ainsi que de toute autre relation exigée.

B.3.1.5 Exemple

networkCTP

Description non formalisée

DEFINITION

"L'objet d'information networkCTP représente une extrémité d'une connexion linkConnection."

Description semi-formalisée

networkCTP INFORMATION OBJECT CLASS

DERIVED FROM networkInformationTop;

CHARACTERIZED BY

networkCTPPackage PACKAGE

BEHAVIOUR

networkCTPPackageBehaviour BEHAVIOUR

DEFINED AS

"<DEFINITION>";;;;

Description formelle

networkCTP_Static

networkCTP : F OBJECT

networkInformationTop_Static

networkCTP \subseteq networkInformationTop

networkCTP_Dynamic

Δ networkCTP_Static

networkInformationTop_Dynamic

Relations potentielles

<clientServer>

<extremitiesTerminateTransportEntity>

<networkTTPAdaptsNetworkCTP>

<subnetworkTPIsRelatedToExtremity>

B.3.2 Relations entre informations

B.3.2.1 Description non formalisée

La partie non formalisée de la spécification de la classe de relations est structurée en quatre parties:

- la première, débutant par le mot clé DEFINITION, contient une définition de la classe de relation;
- la deuxième, débutant par le mot clé ROLE, décrit les rôles de la classe de relations;
- la troisième, débutant par le mot clé INVARIANT, indique les invariants susceptibles de s'appliquer aux objets d'information jouant des rôles dans la relation pendant la durée de vie de cette dernière. Ces invariants peuvent être exprimés au moyen d'un langage non formalisé, par exemple inv_2: "le conteneur et les éléments doivent avoir le même attribut de direction";
- la dernière partie, débutant par le mot clé TRANSITION, indique des limitations pour les transitions d'état des objets d'information jouant des rôles dans la relation.

B.3.2.2 Description semi-formalisée

La partie semi-formalisée de la spécification de la relation entre informations est rédigée en utilisant une notation GRM étendue. L'extension faite à la directive GRM est constituée par la modification de la clause COMPATIBLE WITH qui prend en charge plus d'une étiquette de classe³. La contrainte de ne faire référence au plus qu'à un seul objet dans la clause COMPATIBLE WITH conduit à une duplication des relations. Cette extension de la notation GRM contribue à améliorer la lisibilité d'un document.

L'expression GRM BEHAVIOUR sera structurée d'une manière correspondant aux parties de la description non formalisée comme suit:

- la partie définition est référencée par l'inclusion du mot clé DEFINITION;
- la partie INVARIANT est copiée dans l'expression BEHAVIOUR, à l'exception de ceux des invariants concernant les cardinalités qui sont indiquées dans l'expression PERMITTED-ROLE-CARDINALITY-CONSTRAINT;
- la partie TRANSITION est copiée dans l'expression BEHAVIOUR, à l'exception des invariants qui concernent l'évolution dynamique des relations impliquant le départ ou l'arrivée d'instances d'objets qui ont été traduits au moyen des expressions BIND-SUPPORT et UNBIND-SUPPORT.

La partie ROLE de la spécification non formalisée est en outre traduite dans l'expression ROLE incluant l'expression COMPATIBLE WITH.

B.3.2.3 Description formelle

La description formelle utilise la notation Z. L'Appendice VII donne un exemple d'application de la notation Z pour cette description.

B.3.2.4 Exemple

linkHasLinkConnections

Description non formalisée

DEFINITION

"La classe de relations linkHasLinkConnections décrit la relation existant entre un lien et les connexions linkConnections qui en font partie.
Ce type de relation est un sous-type de setOf."

ROLE

container

"Joué par une instance du type ou sous-type d'objet d'information lien."

element

"Joué par une instance du type ou sous-type d'objet d'information linkConnection."

Description semi-formalisée

linkHasLinkConnections RELATIONSHIP CLASS

DERIVED FROM setOf;

BEHAVIOUR

linkHasLinkConnectionsBehaviour BEHAVIOUR

DEFINED AS

"<DEFINITION>";;

³ Il est possible de définir une classe d'objets gérés dérivée d'un supertype commun de la liste des classe d'objets, avec une classe de comportement stipulant que ces instances sont compatibles avec les classe d'objets de la liste. Ce procédé peut être utilisé pour la définition GRM effective dans la perspective entreprise.

ROLE container

COMPATIBLE WITH link AND SUBCLASSES;

ROLE element

COMPATIBLE WITH linkConnection AND SUBCLASSES;

Description formelle

linkHasLinkConnections_Static

linkHasLinkConnections : F RELATIONSHIP

setOf_Static

link_Static

linkConnection_Static

linkHasLinkConnections \subseteq setOf

$\forall R : \text{linkHasLinkConnections} \bullet \text{container}(R) \in \text{link} \wedge \text{elementSet}(R) \in \text{linkConnection}$

linkHasLinkConnections_Dynamic

Δ linkHasLinkConnections_Static

setOf_Dynamic

link_Dynamic

linkConnection_Dynamic

B.3.3 Définition du schéma statique

Le schéma statique est utilisé pour définir un ensemble d'états d'un système à un instant donné. Il définit des états globaux et génériques du système, c'est-à-dire des états qui impliquent un ou plusieurs attributs appartenant à un ou plusieurs objets d'information. Ceci est nécessaire pour la description de l'état du système avant une transition globale (préconditions d'une opération informatique) et l'état du système après cette transition (postconditions)⁴.

L'état d'un système est exprimé selon les règles suivantes:

- contraintes de valeur d'attribut (au sein d'un objet ou entre objets au moyen des contraintes de relation);
- existence⁵ d'objets.

Il n'est nécessaire de spécifier que ceux des schémas qui ont un intérêt.

B.3.3.1 Description non formelle

La partie non formalisée de la spécification de la classe d'objet est structurée en trois parties:

- la première partie, débutant par le mot clé DEFINITION donne la sémantique de l'état global;
- la deuxième, débutant par le mot clé ROLE, donne la liste de tous les rôles pour lesquels est défini une contrainte ou un invariant;
- la troisième, débutant par le mot clé INVARIANT, donne la liste des règles définissant l'état.

⁴ Bien qu'elles fassent partie de la perspective informationnelle, les transitions entre ces états seront décrites dans la perspective informatique.

⁵ Il convient de noter que cette clause peut être exprimée au moyen de la règle précédente en tenant compte du fait qu'un objet n'existe que si, et seulement si, son attribut identificateur d'objet n'est pas nul.

B.3.3.2 Description semi-formalisée

La définition de la méthode de spécification utilisée pour la description semi-formalisée appelle une étude ultérieure. L'utilisation de l'expression RELATIONSHIP CLASS de la directive GRM a été proposée comme méthode de spécification.

B.3.3.3 Description formelle

La définition de la méthode de spécification utilisée pour la description formelle appelle une étude ultérieure. Il a été proposé d'utiliser la notation Z.

B.3.3.4 Exemple

ssccNotConnected

Ce concept d'information est lié aux entités entreprise suivantes:

```
<"Rec. G.852.1",COMMUNITY:sscc,ACTION:sscc1,OBLIGATION:OBLG_2>,  
<"Rec. G.852.1",COMMUNITY:sscc,ACTION:sscc1,OBLIGATION:PROH_1>,  
<"Rec. G.852.1",COMMUNITY:sscc,ACTION:sscc2,OBLIGATION:OBLG_2>.
```

Description non formalisée

DEFINITION

"Le schéma sccNotConnected définit un type de schéma contenant deux sous-types d'objets subnetworkTPinformation non connectés qui sont candidats au service de gestion de connexion point à point."

ROLE

involvedSubnetwork

"Joué par une instance du type ou sous-type d'objet d'information sccSubnetwork."

potentialAEnd

"Joué par une instance des types ou sous-types d'objets d'information sccSubnetworkTPSink, sccSubnetworkTPSource ou sccSubnetworkTTPBidirectional."

potentialZEnd

"Joué par une instance des types ou sous-types d'objets d'information sccSubnetworkTPSink, sccSubnetworkTPSource ou sccSubnetworkTTPBidirectional."

INVARIANT

inv_1

"Les objets jouant les rôles potentialAEnd and potentialZEnd rôles sont impliqués dans une instance du type de relation subnetworkIsDelimitedBy avec l'objet jouant le rôle involvedSubnetwork."

inv_2

"L'objet jouant le rôle potentialAEnd rôle n'est pas impliqué dans une instance du type ou sous-type de relation subnetworkConnectionIsTerminatedByPointToPoint relationship."

inv_3

"L'objet jouant le rôle potentialZEnd rôle n'est pas impliqué dans une instance du type ou sous-type de relation subnetworkConnectionIsTerminatedByPointToPoint relationship."

B.3.4 Schémas dynamiques

Un schéma dynamique est utilisé pour exprimer les transitions valides entre deux ou plusieurs schémas statiques.

B.3.4.1 Description non formalisée

Un ensemble de définitions de schéma statique peut être fourni comme précondition pour la spécification d'un ensemble de schémas statiques constituant des postconditions. La spécification non formalisée sera faite dans le format suivant:

Label DEFINITION "text"

PRE_CONDITION <static_schema_label> ou "text string"

POST_CONDITION <static_schema_label> ou "text string"

dans lequel la chaîne "text string" ne sera utilisée que pour exprimer des contraintes sur des valeurs d'attribut d'un objet unique.

B.3.4.2 Description semi-formalisée

Appelle une étude ultérieure.

B.3.4.3 Description formelle

Appelle une étude ultérieure.

B.3.5 Attributs

B.3.5.1 Description non formalisée

La partie non formalisée de la définition d'attribut est structurée en quatre parties:

- la première partie, débutant par le mot clé DEFINITION, contient une définition de l'attribut;
- la deuxième partie, débutant par le mot clé STATE, donne la liste des valeurs pouvant être prises par un attribut avec la sémantique correspondante;
- la troisième partie, débutant par le mot clé INVARIANT donne la liste éventuelle de tous les invariants valables pour cet attribut;
- la quatrième partie, débutant par le mot clé TRANSITION, donne la liste de toutes les transitions possibles entre les états de l'attribut. Ceci peut être fait par la définition d'un tableau de transition.

Les définitions d'attributs donnent la sémantique des valeurs, mais non leur syntaxe. Si les définitions d'attributs sont compatibles avec des attributs appartenant à des modèles d'objets gérés GDMO existants (par exemple dans la Recommandation G.774), alors la référence de ces attributs sera fournie. Dans ce cas, la spécification de perspective informationnelle importe la sémantique d'un attribut, mais non sa syntaxe (qui peut être importée dans la perspective informatique correspondante).

B.3.5.2 Description semi-formalisée

La description semi-formalisée utilise le squelette ATTRIBUTE de la directive GDMO sans la clause WITH ATTRIBUTE SYNTAX.

B.3.5.3 Description formelle

La description formelle utilise la notation Z. L'Appendice VII donne un exemple d'application de la notation Z pour cette description.

B.3.5.4 Exemple

userLabel

Description non formalisée

DEFINITION

"Le type d'attribut userLabel attribue un nom convivial à la ressource correspondante. La sémantique de cet attribut est importée de l'attribut M:3100: 1994 userLabel."

Description semi-formalisée

**userLabel ATTRIBUTE
BEHAVIOUR
DEFINED AS
"DEFINITION";**

Description formelle

[UserLabel]

userLabel_Static

userLabel: OBJECT F UserLabel

userLabel_Static

Δ userLabel_Static

\forall object : OBJECT | object \in **dom** userLabel \cup **dom** userLabel' •
userLabel'(object) =userLabel(object)

ANNEXE C

Description du squelette de la perspective informatique

C.1 Introduction

Les squelettes définis dans la présente annexe doivent être mappés vers les squelettes utilisés pour le domaine de communication dans la mise en œuvre d'ingénierie. Ces squelettes sont définis au moyen d'un sous-ensemble de la syntaxe abstraite ASN.1. La représentation faite dans cette syntaxe abstraite et les opérations informatiques doivent être mappées vers la syntaxe particulière et le protocole du domaine de communication utilisés dans la mise en œuvre d'ingénierie. Ce processus aboutit à une spécification informatique qui est propre à chaque domaine de communication envisagé.

Une interface peut avoir de multiples opérations, chacune d'elles étant définie au moyen d'un squelette d'opération.

Le squelette d'objet est destiné à la classe d'objets informatiques. Toute instance de la classe peut être présente de multiples fois pour toute interface donnée.

Le langage de définition utilisé est la notation ASN.1. Les limitations suivantes sont recommandées afin de faciliter la traduction des définitions des langages de type IDL/ODL.

- utiliser SEQUENCE OF au lieu de SET OF
- éviter l'utilisation de ANY ou de ANY DEFINED BY
- limiter l'utilisation de définitions récursives de type à:
 - name1 ::= SEQUENCE {
.
.
.
SEQUENCE OF name1}
 - avec name1 = name1

NOTE – Cette expression prend en charge le filtre CMIS.

C.2 Directives générales

Le présent sous-paragraphe fournit des directives générales concernant l'habillage des squelettes dont la définition formelle est donnée au C.3. Elle fournit une information concernant le mappage vers des entités d'information de la perspective informationnelle.

Dans la description d'une opération faite dans la perspective informatique, les préconditions et postconditions de l'invocation d'une opération peuvent être définies directement comme invariants dans la clause de comportement du squelette d'opération, ou elles peuvent être spécifiées par des références aux schémas définis dans la perspective informationnelle.

Ces schémas peuvent faire l'objet de références afin de définir les préconditions qui doivent être présentes dans le système avant que ces opérations puissent être exécutées et les postconditions auxquelles le système doit se conformer après l'exécution. Le schéma statique spécifie un ensemble de systèmes en fournissant:

- les objets impliqués dans toutes les instances de la classe système (en utilisant la clause <role> dans la clause de relation d'information de la perspective informationnelle);
- des contraintes sur leurs états (en utilisant la clause <invariant>);
- des contraintes sur les états combinés décrits dans des relations (en utilisant la clause <invariant>).

Le paramètre de l'opération sélectionnera quelle instance de système sera adressée dans la précondition et quelle instance de système sera fournie comme résultat de l'opération dans la postcondition. Le comportement peut contenir en conséquence:

- une clause précondition qui peut faire référence à un squelette de la spécification de la perspective informationnelle;
- une clause postcondition qui peut faire référence à un squelette de la spécification de la perspective informationnelle;
- une clause règle de comparaison de paramètre ou un texte donnant une information à propos de cette sélection.

Une instance unique de type interface informatique ne peut prendre en charge à la fois les interfaces client et serveur. De même elle ne peut pas prendre en charge à la fois les clauses PRE/POST CONDITIONS et TRIGGERING CONDITION.

C.2.1 Directives générales propres au squelette d'opération

Ce sous-paragraphe fournit des directives au sujet du contenu des entités dans un squelette OPERATION.

C.2.1.1 Squelette de paramètre

Un paramètre d'opération informatique est associé à un objet d'information. Un type lui est associé. Un paramètre doit être défini avec une clause d'adéquation de paramètre.

C.2.1.1.1 Paramètres d'exploitation servant à identifier les objets

Dans la perspective informatique, les interfaces sont définies de sorte à permettre l'accès à l'abstraction (perspective informatique) d'une ressource qui est identifiée dans la perspective entreprise. Chaque interface informatique fournit un accès à une fin de gestion bien particulière. Dans la perspective informatique, la ressource d'entreprise adressée peut être identifiée par la référence à l'une quelconque de ses interfaces. La seule restriction réside dans le fait que l'interface informatique servant à identifier une ressource doit toujours être présente sur la ressource. Par exemple, la référence à un port dans la communauté d'entreprise ssc peut se faire par l'utilisation

d'une référence à ses interfaces `ssccSnTpIfce` associées dans la perspective informatique correspondante.

Les types d'interface informatique, qui sont indépendants du domaine de communication, sont mis en correspondance avec des types d'interface qui dépendent du domaine de communication (ou des "objets") dans la perspective ingénierie. Par exemple, lors de la mise en correspondance avec le domaine de communication de gestion OSI, chaque interface peut être mise en correspondance avec une classe d'objets gérés ou avec un paquetage d'une classe d'objets.

Selon la troisième partie du modèle RM-ODP, "un paramètre formel qui est un identificateur d'interface informatique est qualifié par un type de signature d'interface informatique. Le paramètre réel correspondant doit faire référence à une interface ayant ce type de signature d'interface (ou l'un de ses sous-types). Le paramètre réel peut uniquement être utilisé comme s'il référençait une interface informatique avec le même type de signature que le paramètre formel (ou l'un des supertypes du paramètre formel)."

Par conséquent, lors du passage d'un paramètre dans une opération, le type de ce paramètre doit être le même que le type de l'interface informatique à laquelle il est fait référence dans la définition de l'opération. Dans les squelettes indépendants du domaine de communication, la référence à une interface typée sert à indiquer un paramètre de référence d'interface dans les signatures d'opération informatique. La notation:

```
"<param_label> : <type> ::= (<interface_type_name>)"
```

sert à spécifier un paramètre qui est une référence à une instance d'interface conforme au type d'interface qui s'appelle `<interface_type_name>`.

Citons en exemple:

INPUT_PARAMETERS

```
snpa : SnTPID ::= (ssccSnTpIfce);
```

où:

snpa -- indique que le nom du paramètre d'entrée est `snpa`

: SnTPID -- indique que le nom du type de paramètre est `SnTPID`

::= (ssccSnTpIfce) -- indique que le nom `SnTPID` est de type `ssccSnTpIfce`, qui peut être satisfait par une interface `ssccSnTpIfce` ou l'un de ses sous-types.

Lorsque la référence à une interface sert de paramètre dans une opération informatique, la signature de cette opération informatique doit indiquer le type d'interface qui est passé sous forme de paramètre réel au moment de l'exécution. Par conséquent, le paramètre réel passé doit contenir l'identité d'une instance d'objet qui représente le type d'interface défini (ou l'un de ses sous-types), au moment de l'invocation pendant l'exécution, dans une réalisation d'ingénierie de l'opération.

Au moment de la spécification, la référence à une interface utilisée sous forme de paramètre doit, par convention, définir de manière non ambiguë le type du paramètre (formel) en question.

NOTE – La valeur d'un paramètre de référence d'interface peut servir ultérieurement à se rattacher à l'instance d'interface associée pour invoquer des opérations à manipuler ou pour interroger l'état d'une ressource.

Le mappage de l'identificateur d'une interface, dans des domaines de communication spécifiques, peut être réalisée de manière plus appropriée pour le mappage de chaque type d'interface avec le type sous-jacent associé au mécanisme d'accès d'ingénierie (par exemple, nom de l'instance d'objet géré ou référence à un objet CORBA).

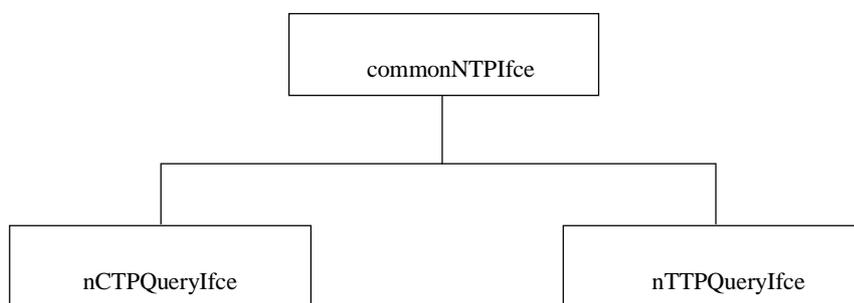
C.2.1.1.2 Types de paramètres informatiques

La définition donnée en notation BNF au C.3 définit la spécification de syntaxe d'un paramètre informatique soit comme une spécification d'un type ASN.1 unique, soit comme une spécification de référence d'interface.

La spécification de type ASN.1 utilise soit une production de type ASN.1 en ligne soit une référence de production de type ASN.1.

Une spécification de référence d'interface utilise le mot clé "REF" (par exemple REF [<commonNTPIfce >]).

Le type d'interface de niveau le plus élevé dans la hiérarchie des interfaces est utilisé en général comme type d'interface nommé dans la spécification REF et il fait implicitement référence à une instance d'interface de ce type ou de tout sous-type compatible créé par héritage. Le nom de l'interface dans la spécification REF peut être remplacé au moment de l'exécution par toute instance d'interface d'un type dérivé de l'interface nommée de la référence.



T1521090-96

Figure C.1/G.851.1 – Exemple de hiérarchie d'interface

La hiérarchie simplifiée donnée en exemple par la Figure C.1 présente un type d'interface "commonNTPIfce" qui est un supertype commun aux deux interfaces dérivées "nCTPQueryIfce" et "nTTPQueryIfce". Les opérations de ce supertype figurent par héritage dans les interfaces dérivées. Il s'ensuit qu'une référence à l'une ou l'autre des interfaces dérivées est un substitut convenable à une référence à l'interface du supertype. Toutefois une référence à l'interface du supertype (commonNTPIfce) ne pourra pas nécessairement servir de référence pour une instance de l'un des types d'interfaces dérivés.

C.2.1.2 Squelette d'adéquation de paramètre

Une clause d'adéquation de paramètre spécifie l'ensemble d'objets d'information ou d'attributs qui doivent être liés au paramètre. Elle spécifie un objet d'information soit directement, soit comme RÔLE joué dans une relation entre informations, soit encore comme une valeur d'attribut d'un objet d'information. La clause d'adéquation limite le paramètre acceptable d'une manière plus stricte que la déclaration de type dans la déclaration du paramètre. Le terme facultatif "ELEMENTS" sert à indiquer que l'expression d'adéquation est appliquée à chaque élément d'un paramètre composite.

C.2.1.3 Préconditions et post conditions

Les préconditions et postconditions dans l'invocation de l'opération sont spécifiées directement sous la forme d'invariants dans la clause comportement de la perspective informationnelle, ou peuvent être spécifiées en faisant référence à un schéma défini dans la perspective informationnelle. Ces invariants définissent les conditions auxquelles doit satisfaire un système pour que ces opérations

puissent être exécutées et les postconditions sont énoncées pour garantir que l'opération s'est achevée correctement.

C.2.1.4 Exceptions

Les exceptions sont activées en fonction des invariants des préconditions et postconditions et spécifient les actions à prendre si l'évaluation d'un invariant donne un résultat "Faux". Comme il n'existe pas d'environnement d'exécution implicite, toutes les exceptions doivent être spécifiées d'une manière explicite. Un invariant faisant référence à la règle d'adéquation du paramètre doit être fourni pour signaler une exception correspondant à une erreur d'adéquation de paramètre.

C.2.2 Définitions d'interface client serveur

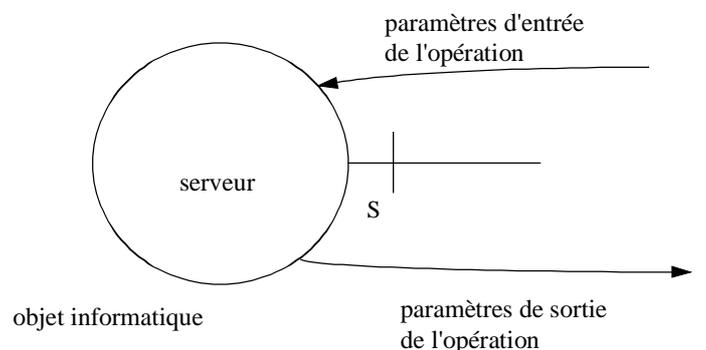
Les objets informatiques peuvent interagir par l'intermédiaire d'une connexion établie entre une interface client et une interface serveur. Une interface informatique peut être une interface d'opération ou une interface de notification. Les rôles de ces interfaces relativement aux types d'interface sont les suivants:

- un rôle de client d'opération: pouvant invoquer des opérations sur des interfaces serveur d'opération;
- un rôle de serveur d'opération: recevant des opérations en provenance d'interfaces client d'opération;
- un rôle de client de notification: invoquant des notifications sur des interfaces de serveur de notification.
- un rôle de serveur de notification: recevant des notifications en provenances d'interfaces client de notification.

Les opérations sont émises par une interface client d'opération et reçues par une interface serveur d'opération. Les notifications sont émises par une interface client de notification et reçues par une interface serveur de notification.

Dans le cas d'une interface serveur d'opération, les paramètres d'entrée contiennent l'information de demande en provenance de l'interface client d'opération. Voir la Figure C.2.

Dans le cas d'une interface client de notification, les paramètres d'entrée contiennent le contenu effectif de la notification. Voir la Figure C.3. Pour cette raison, le contenu effectif de la notification est spécifié sous forme de paramètres d'entrée de l'opération définissant la notification.



T1521100-96

Figure C.2/G.851.1 – Interfaces d'opération de serveur pour des objets informatiques

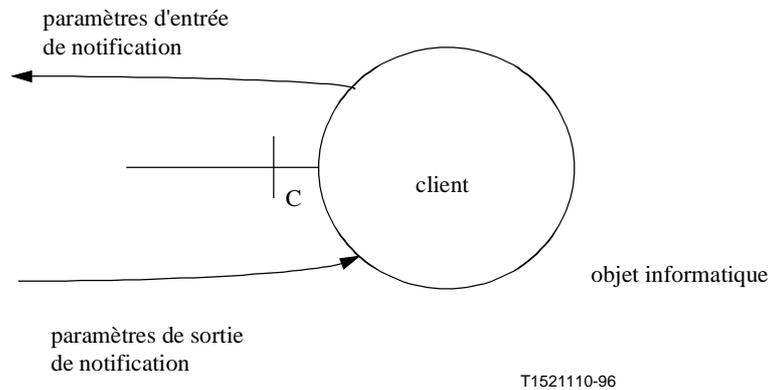


Figure C.3/G.851.1 – Interface client de notification pour des objets informatiques

C.2.3 Mappage vers divers domaines de communication

Les interfaces informatiques indépendantes du domaine ne sont pas toutes obtenues par héritage à partir d'un objet "sommital" commun. Il en résulte que le mappage des squelettes informatiques indépendants du domaine vers la représentation GDMO, des définitions d'interface qui ne sont obtenues par héritage doivent être remplacées par une définition d'objet géré héritée de la classe "sommitale" d'objets gérés GDMO. Toute spécification REF peut être traduite sous forme d'un type ASN.1 hérité de la classe "ObjectInstance", puisque les solutions d'ingénierie CMIP utilisent les noms d'objets gérés pour faire référence aux objets gérés des interfaces serveur.

Le mappage des squelettes de l'interface informatique indépendants du domaine de communication vers les définitions d'interface spécifiques du domaine de communication (par exemple le langage IDL CORBA ou l'élément GDMO/CMISE) pour des mises en œuvre d'ingénierie particulières est fait en général par insertion d'un objet "sommital" adéquat au sommet de la hiérarchie d'interfaces. En outre, le type REF est mappé vers un type adéquat fournissant la référence des interfaces.

Les attributs de la classe d'objet géré GDMO "sommital" permettent à un client de déterminer la classe qui a effectivement servi à instancier un objet géré. Un client peut en particulier invoquer une primitive CMIP M-get pour l'attribut managedObjectClass, avec le paramètre CMIP d'instance d'objet géré positionné sur la valeur de la référence et le paramètre de classe d'objet géré positionné sur la valeur "Recommandation X.721"::actualClass value [12].

De même le langage IDL CORBA possède une interface commune avec un supertype dénommé "Object" qui possède une opération appelée "narrow" permettant à un client de déterminer avec quels types la référence effectivement instanciée est compatible. Comme le langage IDL CORBA possède un type pour l'interface de référence, la spécification REF peut effectuer directement un mappage vers la référence d'interface CORBA au moyen de mappages vers les squelettes d'interfaces spécifiques du domaine CORBA.

C.3 Définitions formelles de squelette

Le présent sous-paragraphe définit les squelettes informatiques au moyen la notation BNF.

C.3.1 Squelette de classe d'objets

```

<computational_object_template> ::= <computational_object_header>
                                     "{"<computational_object_body>"}"
<computational_object_header> ::=  "COMPUTATIONAL_OBJECT_CLASS
<object_name>

```

```

<object_name> ::=      <identifiant>

<computational_object_body> ::=      [<server_interface_definitions>]
                                       [<client_interface_definitions>]
                                       [<behaviour_definition>]

<server_interface_definitions> ::=      "SERVER_INTERFACES"
                                       {<server_interface_label> ";" }*

<server_interface_label> ::=      <label_reference>

<client_interface_definitions> ::=      "CLIENT_INTERFACES"
                                       {<client_interface_label> ";" }*

<client_interface_label> ::=      <label_reference>

<behaviour_definition> ::=      "BEHAVIOUR" {<text_delimiter> <string_literal>
                                       <text_delimiter>
                                       <string_literal> }";"

```

```

<identifiant> ::= [a-zA-Z][-a-zA-Z0-9_:.]*

```

-- Cette définition accepte les identifiants commençant par une lettre et contenant
-- des lettres, des chiffres, des soulignements, des tirets, des deux points et des points.

```

<text_delimiter> ::=      ! | " | # | $ | % | ^ | & | * | ' | ' | ~ | ? | @ | \

```

NOTE – Si un délimiteur de texte est utilisé, le même caractère sera utilisé en début et en fin de chaîne et chaque fois que le caractère délimiteur de texte apparaît au sein de la chaîne, il sera remplacé par une double occurrence de ce caractère. Si aucun caractère délimiteur de texte n'est utilisé, la chaîne ne contiendra aucun des caractères de ponctuation qui sont des successeurs valides à une chaîne de texte du squelette BEHAVIOUR (c'est-à-dire ";").

C.3.2 Squelette d'interface

```

<computational_interface_template> ::=      <computational_interface_header>
                                       "{" <computational_interface_body> "}"

<computational_interface_header> ::=      "COMPUTATIONAL_INTERFACE"
                                       <interface_name>

<interface_name> ::=      <identifiant>

<computational_interface_body> ::=      ["DERIVED FROM" <interface_label>]
                                       <operation_definitions>
                                       [<behaviour_definition>]

<interface_label> ::=      <server_interface_label> | <client_interface_label>

<operation_definitions> ::=      "OPERATION"
                                       {<operation_label> ";" }*

<operation_label> ::=      <label_reference>

```

C.3.3 Squelette d'opération

```
<operation_template> ::=          <operation_header> "{" <operation_body> "}"

<operation_header> ::=          "OPERATION" <operation_name>

<operation_name> ::=           <identifiant>

<operation_body> ::=
    {INPUT_PARAMETERS" [{"<param_label> ":" <syntax_label> ";"}* ]}
    [{"OUTPUT_PARAMETERS" [{"<param_label> ":" <syntax_label> ";"}*]}]
    [{"RAISED_EXCEPTIONS" [{"<exception_label>":" <syntax_label>";"}*]}]
    <opn_behaviour_definition> ";"]

<param_label> ::=             <identifiant>

<exemption_label> ::=        <identifiant>

<syntax_label> ::=           <primitive_asn1_type_name> | <module_name> "::" <production_name>
    <type_production> | <production_name>

<production_name ::=         <identifiant>
    <module_name> ::= <identifiant>

<primitive_asn1_type_name> ::= <identifiant>

<asn1_production> ::=        <type> "::=" <comp_type_def>

<comp_type_def> ::=         <interface_reference> | <singleASN1typedef>
    -- Importé de la Recommandation X.208

<ref_spec> ::=             <identifiant>

<opn_behaviour_definition> ::= "BEHAVIOUR"
    [{"INFORMAL" [{"<text_delimiter> <string_literal> <text_delimiter>
        |<string_literal> ";"}]}]
    "SEMI-FORMAL"
    {"PARAMETER_MATCHING"
    {<param_label> [{"ELEMENTS"}] ":" <parameter_matching_expression> ";"}*}
    [{"{"PRE_CONDITIONS" [{"<schema_label> | {<text_delimiter>
        <string_literal> <text_delimiter>"};"}]}]
    [{"{"POST_CONDITIONS" [{"<schema_label> | {<text_delimiter>
        <string_literal> <text_delimiter>}";"}]}] |
    {"TRIGGERING_CONDITIONS
    [{"<transition_label> | <text_string> |
        <state_label> TRANSITION_TO <state_label> ";"}]}]

    [{"EXCEPTIONS
    [{"IF <exception_invariant_label> "NOT_VERIFIED"
        "RAISE_EXCEPTION" <exception_label> ";"}*]}]

<schema_label> ::=          <label_reference>

<invariant_label> ::=       <label_reference>

<exception_invariant_label> ::= {"PRE_CONDITION" <label_reference>}
    | {"POST_CONDITION" <label_reference>}

<transition_label> ::=      <label_reference>
```

```

<text_string> ::=          <string_literal>

<type> ::=    <identifiant>

<string_literal> ::= [a-zA-Z] [-a-zA-Z0-9_:.]*

<state_label> ::=    <label_reference>

parameter_matching_expression ::=
    <label_reference>
    |<parameter_matching_expression> "AND" <parameter_matching_expression>
    |<parameter_matching_expression> "OR" <parameter_matching_expression>
    |"NOT" <parameter_matching_expression>
    |"{" <parameter_matching_expression>"}"

<param_reference> ::=      <label_reference> | "NOT {" <label_reference> }"

```

C.4 Exemple

Cet exemple présente l'interface informatique et un squelette qui fournit l'établissement d'une connexion dans la communauté de gestion de connexion de sous-réseau.

interface d'exécutant de connexion de sous-réseau simple

L'exécutant de sous-réseau simple gère l'établissement et la libération des connexions de sous-réseau.

L'interface d'exécutant de connexion de sous-réseau simple est nécessaire pour répondre aux besoins d'entreprise indiqués dans

```

<"Rec. G.852.1", COMMUNITY:sscc, ACTION:sscc1 > ,
<"Rec. G.852.1", COMMUNITY:sscc, ACTION:sscc1 > .

```

L'interface d'exécutant de connexion de sous-réseau simple fournit la fonctionnalité d'établissement de connexion de base. L'opération `ssccSetupSubnetworkConnection` établit une connexion de sous-réseau et l'opération `ssccReleaseSubnetworkConnection` supprime la connexion de sous-réseau.

```

COMPUTATIONAL_INTERFACE simpleSncPerformerIfce {
    OPERATION    <setupSubnetworkConnection>;

                <ssccReleaseSubnetworkConnection>;
}

```

établissement de connexion de sous-réseau dans la communauté sscc

Cette opération établit une connexion de sous-réseau simple entre une unique extrémité A (point de terminaison de sous-réseau snTP ou de réseau nTP) et une unique extrémité Z (point snTP ou nTP).

```

OPERATION    ssccSetupSubnetworkConnection {

```

INPUT_PARAMETERS

```
subnetwork : SubnetworkId ::= (ssccSnIfce);
```

```
-- Le paramètre de sous-réseau sert à indiquer le sous-réseau par lequel l'exécutant établit
-- les connexions de sous-réseau. Ce paramètre est utilisé, par exemple, lorsqu'un exécutant donné
-- peut établir des connexions de sous-réseau dans de nombreux sous-réseaux. Si l'exécutant est
-- associé à un seul sous-réseau, le paramètre de sous-réseau de cette opération est redondant, il
-- peut être supprimé permettant ainsi une optimisation au niveau ingénierie.
```

```
snpa : SnTPId ::= (snTPIfce);
snpz : SnTPId ::= (snTPIfce);
dir : Directionality;
suppliedUserLabel : UserLabel;
```

-- une chaîne de longueur nulle signifie qu'aucune étiquette n'est fournie
serviceCharacteristics: CharacteristicsId ::= (serviceCharacteristicsIfce);
-- une référence peut être utilisée pour déterminer toute caractéristique
-- éventuelle de qualité de service ou d'acheminement);

OUTPUT_PARAMETERS

newSNC : SNCId ::= (sncIfce);
agreedUserLabel : UserLabel;

RAISED_EXCEPTIONS

invalidTransportServiceCharacteristics: NULL;
incorrectSubnetworkTerminationPoints : SEQUENCE OF SnTPId;
-- la liste contient un seul élément lorsqu'un seul point est incorrect.
subnetworkTerminationPointsConnected : SEQUENCE OF SnTPId;
-- la liste contient un seul élément lorsqu'un seul point de terminaison
-- de sous-réseau reste connecté
failure : Failed;
wrongDirectionality : Directionality;
userLabelInUse : UserLabel;

BEHAVIOUR

INFORMAL

!

Cette opération établit une connexion de sous-réseau entre une certaine extrémité A (point snTP ou nTP) et une certaine extrémité Z (point snTP ou nTP). Les points de terminaison de sous-réseau (snTP) ou de réseau (nTP) devant être connectés sont spécifiés par leur identification explicite.

Le client peut fournir une étiquette d'utilisateur et une seule. Si cette étiquette n'est pas fournie (c'est-à-dire si la chaîne est de longueur nulle), le fournisseur assigne une étiquette d'utilisateur à la connexion.

Une connexion de sous-réseau ne peut être établie que dans l'état "connecté".

Un unique objet de connexion de sous-réseau non subdivisé sera créé, il sera point à point unidirectionnel ou point à point bidirectionnel. Cet objet aura une extrémité A et une extrémité Z.

La connexion de sous-réseau aura la directivité (unidirectionnelle ou bidirectionnelle) spécifiée dans les paramètres d'opération.

Lorsqu'elles sont utilisées, les caractéristiques de service spécifient un ensemble prédéterminé de paramètres de transport que le serveur peut offrir.

Les réponses correspondant à l'opération d'établissement comprennent tous les renseignements sur les motifs lorsque la demande n'a pas pu être satisfaite.

PRE_CONDITIONS

Les points snTP ou nTP doivent exister pour pouvoir réaliser une connexion dans un sous-réseau donné.

Cette opération se soldera par un échec lorsque l'un quelconque des points de terminaison de sous-réseau ou de réseau spécifiés interviendra déjà dans une connexion de sous-réseau. L'anomalie "subnetworkTerminationPointsConnected" sera signalée.

Cette opération se soldera par un échec lorsque les points de terminaison de sous-réseau ou de réseau ne figureront pas dans le domaine du sous-réseau. L'anomalie "incorrectSubnetworkTerminationPoints" sera signalée.

Cette opération se soldera par un échec lorsque les caractéristiques de service demandées ne seront pas prises en charge par l'objet informatique qui exécute l'opération. L'anomalie "invalidTransportServiceCharacteristics" sera signalée.

POST_CONDITIONS

Si l'un quelconque des paramètres d'entrée de la connexion de sous-réseau ne peut pas être satisfait par le serveur, l'opération se soldera par un échec.

Cette opération se soldera par un échec lorsque la valeur de l'étiquette d'utilisateur de la connexion de sous-réseau sera nulle ou lorsqu'elle ne sera pas unique dans le domaine du sous-réseau en question. L'anomalie "userLabelInUse" sera signalée.

!

SEMI_FORMAL

PARAMETER_MATCHING

subnetwork : < sscNotConnected, ROLE:involvedSubnetwork > AND
 < sscConnected, ROLE:involvedSubnetwork >;

snpa : < sscNotConnected, ROLE:potentialAEnd > AND
 < sscConnected, ROLE:connectedAEnd >;

snpz : < sscNotConnected, ROLE: potentialZEnd > AND
 < sscConnected, ROLE:connectedZEnd >;

dir : < sscConnected, ROLE: involvedSubnetwork ,ATTRIBUTE:
 directionality >;

newSNC : < sscConnected, ROLE: involvedSubnetwork >;

suppliedUserLabel : < sscConnected, ROLE:involvedSubnetwork, ATTRIBUTE: userLabel >

OR <> ; -- l'utilisateur n'a pas à fournir de valeur d'étiquette d'utilisateur

agreedUserLabel : < sscConnected, ROLE:involvedSubnetwork, ATTRIBUTE: userLabel >;

serviceCharacteristics : < sscConnected, ROLE:involvedServiceCharacteristics >;

PRE_CONDITIONS < sscNotConnected > ;

-- Le schéma sscNotConnected définit un type de schéma avec deux

-- sous-types d'objets d'information nTP non connectés, candidats

-- au service de gestion de connexion point à point.

POST_CONDITIONS < sscConnected > ;

-- Le schéma sscConnected définit le type de schéma avec deux objets

-- d'information nTP connectés, candidats au service de gestion de

-- connexion point à point.

EXCEPTIONS

IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
 incorrectSubnetworkTerminationPoints;

IF PRE_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION
 subnetworkTerminationPointsConnected ;

IF PRE_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION
 subnetworkTerminationPointsConnected ;

IF POST_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
 failure;

IF POST_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION
 failure;

IF POST_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION
 failure;

IF POST_CONDITION <inv_4> NOT_VERIFIED RAISE_EXCEPTION
 userLabelInUse;

;

}

ANNEXE D

Squelettes et directives générales de la perspective d'ingénierie de la gestion OSI

La présente annexe contient les squelettes utilisés dans la perspective d'ingénierie et fournit également un ensemble de directives générales pour la transformation des objets informatiques et information des objets d'ingénierie GDMO qui sont des membres d'un scénario spécifique.

L'élaboration complète de cette annexe appelle une étude ultérieure.

D.1 Squelettes

Le squelette du modèle Ensemble est utilisé pour définir les objectifs du scénario des objets d'ingénierie.

Le squelette GDMO est utilisé pour la définition des objets d'ingénierie.

D.2 Mappages possibles vers les définitions d'objets gérés

Un objet géré de la perspective entreprise représente un mappage d'objets information et informatiques vers la représentation GDMO, ce qui peut faciliter une implémentation utilisant la gestion OSI et l'élément et le protocole CMISE/CMIP.

Les directives générales pour la création de définitions d'objet gérés sont les suivantes:

- 1) les objets gérés cibles sont construits à partir d'objets d'information et informatiques tels que les suivants:
 - a) attributs en provenance des objets d'information de la perspective informationnelle;
 - b) interfaces informatiques de serveur pouvant être définies comme:
 - des paquetages distincts de la perspective entreprise;
 - ainsi que des objets GDMO distincts (si l'interface du serveur est instanciée).
 - c) opérations appartenant à une interface de serveur définies comme:
 - des opérations GET, GET-REPLACE, REPLACE sur des attributs;
 - des actions définies séparément.
- 2) certaines interfaces client peuvent être réalisées sous la forme d'une interrogation du service d'annuaire X.500 (opérations informatiques exécutant des résolutions nom/adresse, par exemple);
- 3) il est possible de définir une classe d'objets gérés dérivée d'un supertype appartenant à la liste de classes d'objets, dont la classe comportement stipule que ses instances sont compatibles avec les classes d'objet de la liste. Ce procédé peut être utilisé dans la représentation GRM effective dans la spécification informatique spécifique du domaine et dans la perspective entreprise.

ANNEXE E

Syntaxe d'étiquette

E.0 Introduction

La référence d'étiquette se présente sous la forme d'une chaîne de caractère. Un caractère "," dans la chaîne indique que l'entité de l'étiquette située à gauche doit être évaluée avant le traitement du reste de la chaîne. Par exemple dans la chaîne:

<INFORMATION_RELATIONSHIP:compoundLinkHasLinks,ROLE:container, ATTRIBUTE: directionality> est un pointeur vers l'attribut ATTRIBUTE directionality dans l'objet INFORMATION_OBJECT qui joue le RÔLE conteneur dans la relation compoundLinkHasLinks.

E.1 Définition de la syntaxe d'étiquette en notation BNF

```

<label_reference>      ::= "<"<label_string>">"

<label_string> ::= <element_label>
                  | <label_list>
                  | <element_label>","<label_list>

<label_list>         ::= <label_entity>
                  | <label_entity> ","<label_list>

<label_entity>       ::= <clause_label>":"<element_label>

<element_label>      ::= <label_text>
                  -- sans espaces, virgules ou barres obliques
                  -- ou parenthèses angulaires ouvertes.
                  | -- un choix nul indique l'absence d'étiquette pour le mot clé

<clause_label>       ::= <enterprise_clause_label>
                  | <information_clause_label>
                  | <computation_clause_label>
                  | <engineering_clause_label>
                  -- étendu si nécessaire

<enterprise_clause_label> ::=
    | "COMMUNITY"
    | "PURPOSE"
    | "ROLE"
    | "POLICY"
    | "ACTION"
    | "ACTION_POLICY"
    | "ACTIVITY"
    | "ACTIVITY_POLICY"
    | "WITH_ACTION_GRAPH"
    | "PERMISSION"
    | "PROHIBITION"
    | "OBLIGATION"
    | "EXCEPTION"
    | "CONTRACT

    -- voir E.2 pour l'arborescence complète de l'étiquette dans la
    -- perspective entreprise

<information_clause_label> ::=
    | "INFORMATION_OBJECT"
    | "INFORMATION_RELATIONSHIP"
    | "STATIC_SCHEMA"
    | "DYNAMIC_SCHEMA"
    | "ATTRIBUTE"
    | "STATE"
    | "DEFINITION"
    | "INVARIANT"
    | "TRANSITION"
    | "POTENTIAL_RELATIONSHIP"

```

```
| "ROLE"  
| "PRE_CONDITION"  
| "POST_CONDITION"  
| "RELATIONSHIP"
```

*-- Voir E.3 pour l'arborescence complète de l'étiquette dans la
-- perspective informationnelle*

```
<computational_clause_label>::=  
| "COMPUTATIONAL_OBJECT_CLASS"  
| "SERVER_INTERFACES"  
| "CLIENT_INTERFACES"  
| "BEHAVIOUR"  
| "COMPUTATIONAL_INTERFACE"  
| "DERIVED_FROM"  
| "OPERATIONS"  
| "OPERATION"  
| "INPUT_PARAMETERS"  
| "OUTPUT_PARAMETERS"  
| "RAISED_EXCEPTIONS"  
| "PARAMETER_MATCHING"  
| "PRE_CONDITIONS"  
| "POST_CONDITIONS"  
| "TRIGGERING_CONDITIONS"  
| "TRANSITION_TO"  
| "EXCEPTIONS"  
| "NOT_VERIFIED"  
| "RAISE_EXCEPTION"  
| "PRE_CONDITION"  
| "POST_CONDITION"  
| "AND"  
| "OR"  
| "NOT"  
| "INFORMAL"
```

*-- Voir E.4 pour l'arborescence complète de l'étiquette dans la
-- perspective informatique*

```
<engineering_clause_label>
```

*-- Appelle une étude ultérieure –
-- Dépend du domaine de communication sélectionné*

E.2 Structure d'arborescence d'étiquette dans la perspective entreprise

"Rec. G.852.xx"

COMMUNITY

<label>

PURPOSE

ROLE

<label>

PERMISSION

<label>

OBLIGATION

<label>

PROHIBITION

<label>

EXCEPTION

<label>

```

ACTION
  <label>
    PERMISSION
      <label>
    OBLIGATION
      <label>
    PROHIBITION
      <label>
    EXCEPTION
      <label>
ACTIVITY
  <label>
    PERMISSION
      <label>
    OBLIGATION
      <label>
    PROHIBITION
      <label>
    EXCEPTION
      <label>
    EXCEPTION
      <label>
    WITH_ACTION_GRAPH
      <label>
CONTRACT
  <label>

```

E.2.1 Exemples d'utilisation

```

<"Rec. G.852.1",COMMUNITY:sscc,ROLE:caller>
<"Rec. G.852.1",COMMUNITY:sfm,ACTION:sfm3,OBLIGATION:OBLG_1>

```

E.3 Structure de l'arborescence d'étiquette de la perspective informationnelle

```

"Rec. G.853.xx"
  INFORMATION_OBJECT
    <label>
      DEFINITION
      ATTRIBUTE
        <label>
          STATE
            <label>
          INVARIANT
            <label>
          TRANSITION
            <label>
      INVARIANT
        <label>
      TRANSITION
        <label>
      RELATIONSHIP
        <label>
      POTENTIAL_RELATIONSHIP
        <label>
  INFORMATION_RELATIONSHIP
    <label>
      DEFINITION
        <label>
      ROLE
        <label>

```

```

    INVARIANT
      <label>
    TRANSITION
      <label>
  STATIC_SCHEMA
    <label>
    DEFINITION
      <label>
    ROLE
      <label>
    INVARIANT
      <label>
  DYNAMIC_SCHEMA
    <label>
    DEFINITION
      <label>
    PRE_CONDITION
      <label>
    POST_CONDITION
      <label>
  ATTRIBUTE
    <label>
    DEFINITION
      <label>
    STATE
      <label>
    INVARIANT
      <label>
    TRANSITION
      <label>

```

E.3.1 Exemples d'utilisation

```

<"Rec. G.853.1",INFORMATION_OBJECT:networkConnectivity,ATTRIBUTE:signalIdentification>
<"Rec. G.853.2",INFORMATION_OBJECT:monitoredEntity,ATTRIBUTE:operationalStateSTATE: enabled>
<"Rec. G.853.1",INFORMATION_RELATIONSHIP:clientServer,ROLE:client>
<"Rec. G.853.1",INFORMATION_RELATIONSHIP:clientServer,INVARIANT:inv_1>
<"Rec. G.853.2",STATIC_SCHEMA:enabledAndReportOn,INVARIANT:enabled>
<"Rec. G.853.2",DYNAMIC_SCHEMA:reportFailureOnEnabledToDisabled,
PRE_CONDITION:EnabledAndReportOn>
<"Rec. G.853.1",ATTRIBUTE:directionality,STATE:unidirectional>

```

E.4 Structure de l'arborescence d'étiquette dans la perspective informatique

```

"Rec. G.854.xx"
  COMPUTATIONAL_OBJECT_CLASS
    <label>
    SERVER_INTERFACES
      <label>
    CLIENT_INTERFACES
      <label>
  COMPUTATIONAL_INTERFACE
    <label>
    OPERATIONS
      <label>

```

OPERATION
INPUT_PARAMETERS
 <label>
OUTPUT_PARAMETERS
 <label>
RAISED_EXEPTIONS
 <label>
PARAMETER_MATCHING
 <label>
PRE_CONDITIONS
 <label>
POST_CONDITIONS
 <label>
TRIGGERING_CONDITIONS
 <label>
EXCEPTIONS
 <label>

E.4.1 Exemples d'utilisation

```
<"Rec. G.854.1",COMPUTATIONAL_INTERFACE:simpleSncPerformerIfce,
OPERATION:releaseSNC,INPUT_PARAMETER:userLabel>
<"Rec. G.854.1",COMPUTATIONAL_INTERFACE:snQueryIfce,
OPERATION:querySnForSNCs,RAISED_EXCEPTIONS:unconnectedSubnetwork>
```

ANNEXE F

Squelette ensemble du modèle Ensemble

F.1 La technique du modèle Ensemble

La présente annexe décrit de quelle manière le concept et le format du modèle "ensemble" élaboré par forum NM peuvent être utilisés dans la vue du niveau réseau du cadre général du modèle RM-ODP.

Le contenu du modèle Ensemble est constitué, dans la mesure du possible, par des références aux documents de la perspective adéquate. Une partie du texte peut être reproduite dans certains cas pour améliorer la lisibilité.

La description du contenu et le formulaire du squelette se trouvent dans le document du forum de gestion de réseau Forum 025 OMNIPOint 1, The "ensemble" Concepts and Format, August 1992 [8]. (*Format et Concept "ensemble"*).

F.2 Squelette du modèle Ensemble

Le squelette du modèle Ensemble se constitue des parties suivantes:

- introduction;
- contexte de gestion;
- modèle d'information de gestion;
- prescriptions de conformité du modèle Ensemble.

Les sous-paragraphes qui suivent décrivent les différentes parties du modèle Ensemble.

F.2.1 Introduction

L'introduction se constitue d'un texte résumant le scénario de l'application en identifiant les communautés de la perspective entreprise à prendre en charge. Elle se constitue des éléments suivants:

- description globale du problème de gestion utilisant le texte global de la perspective entreprise;
- identification des ressources à gérer;
- besoins globaux à satisfaire et contraintes à respecter;
- identification des interactions entre les interfaces d'ingénierie.

F.2.2 Contexte de gestion

Le présent sous-paragraphe se constitue des éléments suivants:

- description plus détaillée des besoins et contraintes de gestion faite au moyen des règles de gestion énoncées dans la perspective entreprise (par exemple en définissant quelles sont les autorisations et les obligations dans le contexte de gestion donné);
- description des ressources à gérer au moyen de références à la perspective informationnelle;
- définition du protocole d'interface à prendre en charge;
- définition des scénarios dans lesquels les flux de messages à travers les interfaces d'ingénierie sont identifiés et spécifiés au moyen de références aux interfaces informatiques.

F.2.3 Modèle d'information de gestion

Le présent sous-paragraphe fait référence à la perspective d'ingénierie qui a été élaborée pour ce scénario d'application.

F.2.4 Exigences de conformité du modèle Ensemble

Les éléments suivants ont été identifiés:

- prescriptions générales de conformité;
- prise en charge fonctionnelle;
- objets d'ingénierie;
- prescriptions de conformité concernant les solutions d'ingénierie, telles que:
 - objets (par exemple formulaires de conformité MOCS pour les solutions d'ingénierie basées sur les directives GDMO);
 - protocole (par exemple en utilisant le formulaire de conformité PICS);
 - utilisation de services d'annuaire.

Un exemple d'utilisation du concept du modèle Ensemble se trouve dans l'Appendice V.

APPENDICE I

Exemples de squelettes et de directives de spécification

I.1 Comparaison entre les services fournis à l'entreprise et les contrats

Les notions de service et de contrat sont en étroite relation. En fait, les services sont la manifestation externe de l'application du contrat. Les deux concepts peuvent en conséquence être utilisés indifféremment.

Un type de contrat peut être privé ou normalisé. Le fournisseur est responsable de l'établissement et de l'actualisation des contrats privés (seul ou associé à ses clients). Si le contrat est le résultat d'un processus de normalisation, il est établi et actualisé par des organismes de normalisation (par exemple l'UIT-T). Les organismes de normalisation ne peuvent pas établir d'instance de contrats.

Les termes utilisés pour structurer et rédiger les contrats sont très importants car ils seront utilisés pour déterminer les responsabilités en cas de manquement au contrat. Dans ce sens un type de contrat définit des "points de conformité entreprise".

Il est prescrit d'élaborer des types de contrat en partant de types précédents et en leur ajoutant de nouvelles caractéristiques ou en élaborant des caractéristiques composées. Il est utile dans de telles circonstances de procéder de la manière suivante:

- bâtir de nouveaux services en réutilisant des services existants;
- élaborer des contrats privés à partir de contrats normalisés;
- garantir une compatibilité amont en cas d'extensions de service.

Un service composé est constitué de l'union suivante:

- ensemble des caractéristiques définies dans les services qui le composent (appelés services importés);
- caractéristiques définies d'une manière locale.

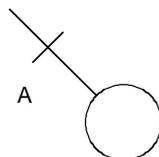
Les deux règles suivantes doivent être respectées en ce qui concerne la combinaison de services:

- les services importés doivent être en harmonie, ce qui signifie qu'ils ne doivent pas comporter de caractéristiques contradictoires;
- les caractéristiques ajoutées d'une manière locale doivent être en harmonie avec les services importés.

APPENDICE II

Représentation d'états combinés

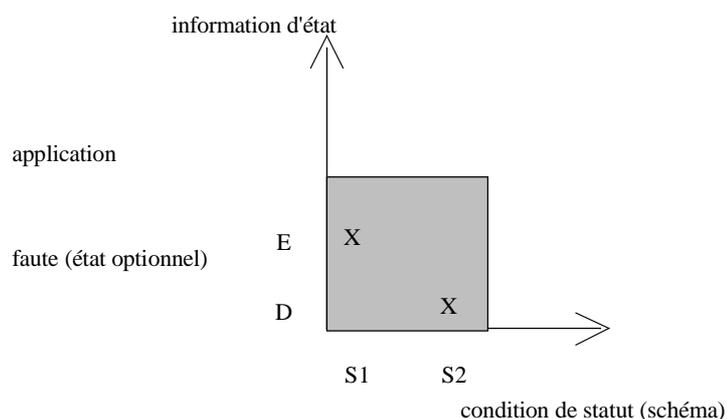
Considérons une application dans laquelle il doit être exprimé si une ressource est en état de fournir un service normal. Cet état peut être vu à travers une interface A vers un objet informatique. Voir la Figure II.1.



T1521120-96

Figure II.1/G.851.1 – Interface informatique

L'état de la ressource vu à travers l'interface A peut être défini par les états "en service" et "ressource défectueuse" (S1, S2). Ceux-ci peuvent être mappés sur les valeurs Activé et Désactivé de l'état opérationnel comme indiqué dans la Figure II.2. Les états S1 et S2 peuvent également être considérés comme constituant un (unique) schéma statique.

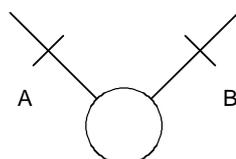


X Valeur d'état spécifiée

T1521130-96

Figure II.2/G.851.1 – Description d'état pour une application en faute

Considérons maintenant la situation décrite par la Figure II.3, dans laquelle l'état de la ressource peut être vu à travers de multiples interfaces. L'interface A reste comme décrit ci-dessus, mais l'interface B est également utilisée pour la fourniture et la configuration d'applications. Le problème consiste à autoriser une application à interfacier l'objet à travers l'interface A qui ne connaît que les états S1 et S2, et également d'autoriser des interactions à travers l'interface B qui possède un domaine d'états plus large.



T1521140-96

Figure II.3/G.851.1 – Interfaces informatiques

La Figure II.4 indique le domaine des états visibles à travers l'interface B. Les états S1 à S6 incluent la table d'états. Tout état est la combinaison des états de base: état administratif et état opérationnel, ainsi que la condition de statut connecté ou non connecté. Les états S3 à S6 peuvent également être définis sous la forme d'un schéma statique.

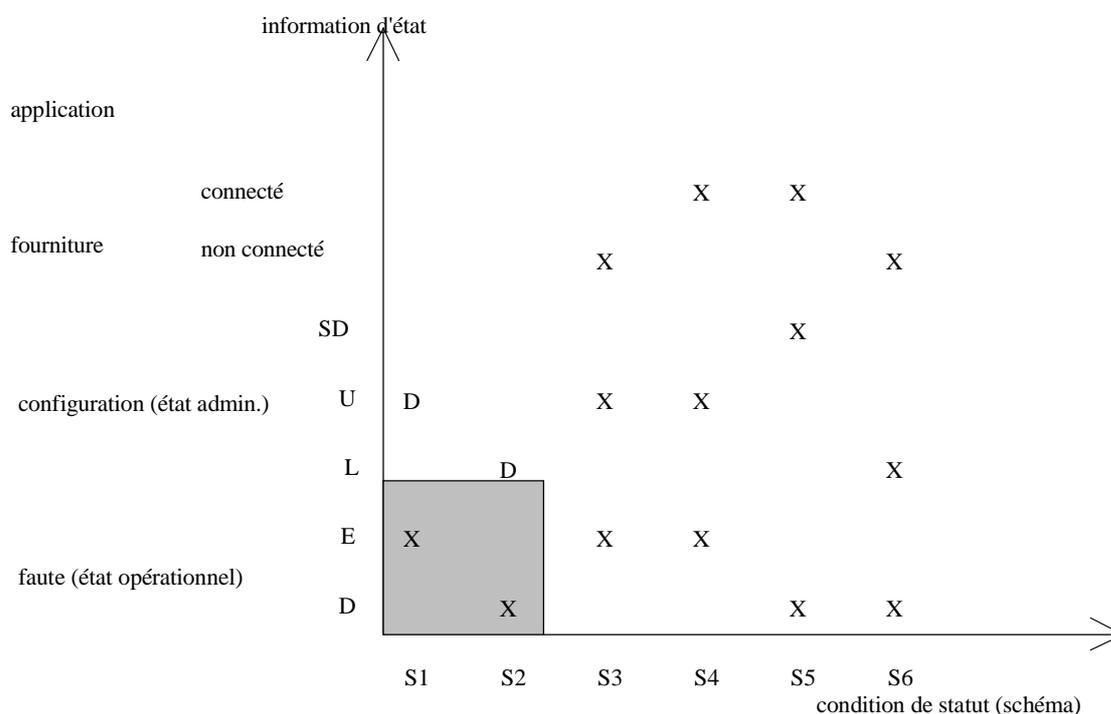
Les états S1 et S2 ont été définis sans connaissance des autres états. Il s'ensuit que des valeurs par défaut sont assignées dans la table d'états de l'interface B aux états de base administratifs et à la condition de statut connecté ou non connecté.

La définition de l'état S5 nécessite trois valeurs d'état de base, alors celle de l'état S1 n'en nécessite qu'une. Si les deux autres états de base de la ressource sont rendus visibles à travers l'interface B, alors le système devra maintenir l'homogénéité des états S1 et S2 avec un contexte d'états plus étendu. On peut considérer que les états supplémentaires se voient attribuer des valeurs par défaut afin de maintenir l'homogénéité avec la table d'état plus étendue. Si les états S1 à S6 sont considérés comme constituant un schéma statique, alors les valeurs S1 et S2 sont inchangées lorsque les valeurs des schémas S3 à S6 sont définies. Le système doit toutefois assurer que les valeurs d'état opérationnel (S1 et S2) fournies à travers l'interface A sont en accord avec les valeurs de l'état opérationnel (de la même ressource) fournies comme états S3 à S6 à travers l'interface B. Dans cet exemple, si les schémas S1, S3 et S4 exigent que l'état opérationnel soit activé et que les schémas S2,

S5 et S6 exigent que cet état opérationnel soit désactivé, alors si S3 et S4 sont fournis à travers l'interface B, S1 doit être fourni à travers l'interface A. De cette manière, les états qui ont été définis à l'origine sur une base d'état simple restent valides lorsqu'ils font partie d'une table beaucoup plus étendue. Un utilisateur de l'interface A ne verra pas de changement lorsque l'objet informatique est étendu afin de prendre en charge l'interface B.

L'ensemble des états dans un objet ou un système peut être étendu en ajoutant de nouveaux états de base et en définissant une relation entre les nouveaux états de base et le schéma existant. L'extension d'état par l'addition d'une nouvelle valeur à un état de base existant doit être évitée. Ceci peut toutefois être nécessaire lorsqu'un mappage doit être fourni du nouvel état de base (étendu) vers l'état de base existant.

Des considérations similaires s'appliquent à la production de deux profils distincts des objets d'information pour deux applications distinctes (et différents objets informatiques).



X valeur d'état spécifiée
D valeur d'état attribuée par défaut

T1521150-96

Figure II.4/G.851.1 – Description d'état pour une application combinée montrant l'utilisation de valeurs par défaut

APPENDICE III

Description de la réalisation du service

III.1 Domaine d'application

La façon dont le fournisseur réalise le service n'a pas d'importance pour le client et ne fait donc pas partie de la spécification du contrat. Le fournisseur doit toutefois documenter, du point de vue de l'entreprise, le mode de réalisation d'un service donné. La réalisation d'un service comprend deux caractéristiques:

- les règles de gestion internes s'appliquant à chaque caractéristique d'un contrat;
- les services ultérieurs qui sont invoqués ensuite comme conséquence de l'application des règles de gestion.

Le fournisseur précédent agira comme client pour ces services ultérieurs. Il est clair que les services ultérieurs peuvent être normalisés ou privés en fonction de la règle de gestion qui régit leur utilisation.

Du point de vue des normes, il pourrait être utile de prévoir un comportement informatif qui régira l'utilisation des services normalisés ultérieurs. Les raisons sont les suivantes:

- justifier l'introduction de nouveaux services normalisés;
- permettre l'expression des contraintes de règles de gestion du service entre des services normalisés liés.

Un fournisseur peut décider, pour diverses raisons, de traiter diverses réalisations de services pour une spécification donnée de service (par exemple en cas de modification des règles de gestion au niveau entreprise). Les clients ne verront pas de modification dans la réalisation d'un service si la spécification de ce dernier n'est pas modifiée.

III.2 Concepts

Alors que le sous-paragraphe précédent peut faire l'objet d'une normalisation, le présent sous-paragraphe fournit une solution pour la réalisation du service. Toute solution n'est évidemment pas unique et peut être modifiée conformément aux règles de gestion du fournisseur du service. Une règle de gestion peut être élaborée pour toute action définie dans le contrat. Dans tout contrat, les organismes de normalisation peuvent, par exemple, fournir une clause de mise en œuvre du service afin d'indiquer de quelle manière les services normalisés peuvent être mis en relation entre eux.

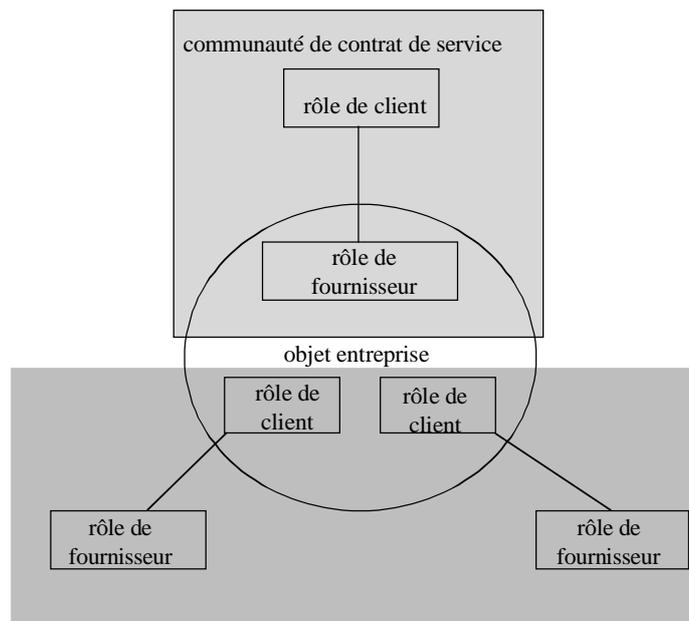
Il est utile dans tous les cas de produire des squelettes normalisés pour la description de la réalisation du service comme partie de la présente Recommandation, de manière à permettre aux fournisseurs du service de décrire leur propre réalisation.

La section de réalisation du service sera structurée en sections subordonnées, à raison d'une par action ou activité définie dans la spécification du contrat. Chacune de ces sections débutera par un élément de communauté indiquant les rôles qui sont impliqués dans l'exécution de l'action ou de l'activité. L'objet entreprise jouant le rôle de fournisseur de service par rapport au contrat du service prendra toujours les rôles d'appelant du service par rapport aux services suivants.

Les éléments de communauté sont suivis par un élément de règle de gestion, chaque règle de gestion étant équivalente à une action.

En conséquence, les règles de gestion de réalisation associées à une action de contrat de service (respectivement à une activité de contrat de service) formeront une description d'activité telle qu'elle sera exprimée sous forme d'un graphe d'actions sans cycle.

Le terme action est utilisé au sens du modèle RM-ODP, c'est-à-dire qu'il peut s'agir d'une action interne ou d'une interaction. Les sections citées précédemment sélectionnent une action interne ou une interaction comme partie de la décision de réalisation. Lorsqu'une action est une interaction, elle sera exprimée sous la forme d'invocations de service, sinon elle sera interne. Voir la Figure III.1.



T1521160-96

Figure III.1/G.851.1 – Exemple de rôle vis-à-vis des objets

Une action n'est pas instantanée et peut être décrite comme:

- en cours: l'action a démarré. Cette action peut être asynchrone et d'autres actions peuvent être réalisées en même temps.
- passée: l'action est terminée. Il s'agit d'une action synchrone et aucune des actions suivantes du graphe ne peut être démarrée avant qu'elle ne soit terminée.
- pouvant se dérouler: cette action est optionnelle et peut être réalisée à la suite de demandes particulières.

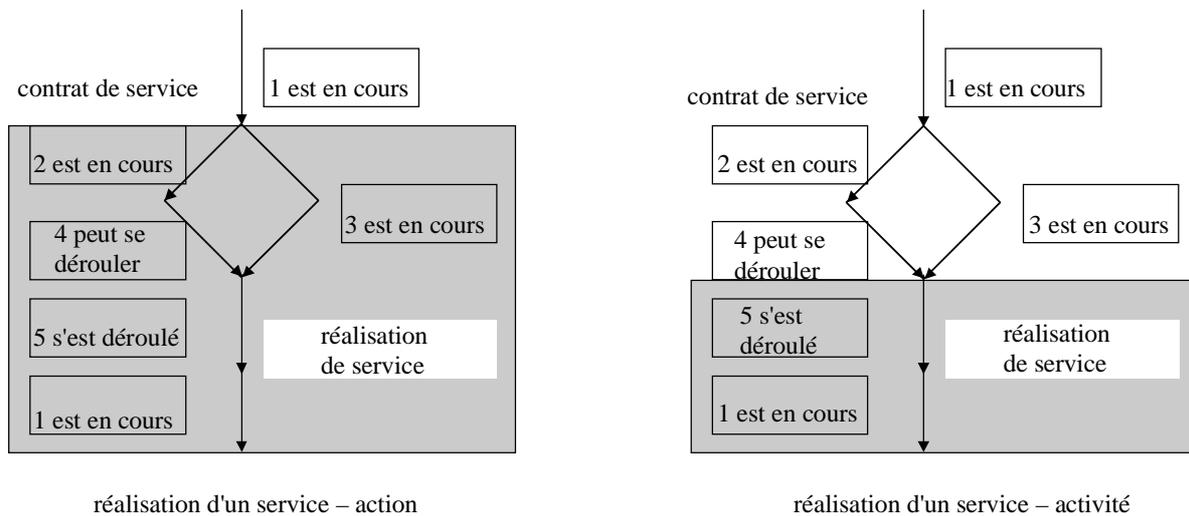
Comme décrit dans la deuxième partie du modèle RM-ODP [2], les actions peuvent se chevaucher dans le temps. Le modèle RM-ODP ne spécifie toutefois pas si des contraintes temporelles existent entre des actions. A des fins de spécification, une action qui va se dérouler ne peut démarrer que si une autre action précédente s'est déroulée.

La description au moyen d'un graphe est spécifiée dans cette sous-section comme un schéma avec des actions pourvues d'étiquettes. Toute action est décrite d'une manière non formalisée en indiquant les points suivants:

- si l'action peut se passer, est en train de se passer ou s'est passée;
- si l'action est interne ou est une interaction (avec le service invoqué);
- en indiquant les raisons du déroulement de cette action à ce niveau du graphe.

Lorsque plusieurs actions apparaissent à un même niveau dans le graphe, elles peuvent se dérouler à tout moment d'une manière indépendante. L'ordre effectif sera spécifié dans d'autres perspectives.

La Figure III.2 donne un exemple de graphe.



T1521170-96

Figure III.2/G.851.1 – Exemple de description d'activité

Squelette

```

{ACTION |ACTIVITY} Name <action label>
Community
  community_definition
Community Roles
  <community label><Role Label>
  role_definition
Community Policies
  Action Name <action label>
  action_definition
  Action Policies
  WITH ACTION GRAPH
  Start
    <action label>
    <nth action label>
  End

```

APPENDICE IV

Exemple d'utilisation du concept et du format du modèle Ensemble

Considérons l'exemple de l'objet d'information `subNetworkConnection` défini dans la Recommandation G.853.1; un objet de ce type représente une connexion de sous-réseau G.805. Des vues partielles des ressources sous-jacentes peuvent être décrites service par service. Les objets `scmSubNetworkConnection` et `msSubNetworkConnectionManagement`, par exemple, constituent respectivement des vues partielles de la connexion de sous-réseau pour le service de gestion de configuration de connexion de sous-réseau (SCM) et pour les services de supervision (MS). Des variables d'état peuvent être définies, dans la spécification d'information, pour chacun de ces types d'objets d'information, sous la forme d'attributs ou de relations.

Une interface informatique est ensuite définie dans la spécification informatique de chaque service. L'interface informatique `sncConfiguration` est définie, par exemple, sous la forme d'une énumération d'opérations pouvant être invoquées sur cette interface (telles que `setupSNCPPointToPoint`, `setupSNCPPointToMultiPoint` et `ReleaseSNC`); il convient de noter que chacune de ces informations

correspond à une action "entreprise" spécifiée dans le contrat de service et pouvant être définie sous la forme suivante:

```
INTERFACE_TEMPLATE sncConfiguration {  
    OPERATION setupSNCPPointToPoint  
    OPERATION setupSNCPPointToMultiPoint  
    OPERATION ReleaseSNC  
    BEHAVIOUR  
    ...  
}
```

Les opérations informatiques sont décrites en termes de signature, de préconditions et de postconditions. Ces deux derniers éléments font référence aux schémas statiques décrits dans la spécification du service. Dans cet exemple, l'opération setupSNCPPointToPoint possède la précondition scmNotConnected et la postcondition scmConnected. La transition entre ces deux états constitue le comportement du système. La manière de réaliser cette transition ou la modalité de respect des invariants n'a pas d'intérêt dans le contexte de spécification normalisée du comportement. Les actions ultérieures éventuelles peuvent soit être considérées comme internes, soit comme interactions avec d'autres objets sur leurs interfaces ouvertes.

Il est nécessaire de définir des profils pour en déduire les interfaces d'ingénierie. La description d'un profil se constitue des quatre parties suivantes:

- des **protocoles de communication** tels que CMISE/CMIP, SNMP, dont le choix aura une influence sur le langage d'ingénierie;
- des **ressources gérées**, par exemple la liste des ressources qui doivent être gérées dans le système (points de terminaison, connexions de sous-réseau, sous-réseaux, etc.);
- des **fonctions**, telles que la gestion de configuration de connexion de sous-réseau et la supervision de point de terminaison de connexion de sous-réseau;
- un **niveau d'abstraction**, par exemple le fait que le profil est défini du point de vue du client ou du point de vue du fournisseur.

Prenons comme exemple une interface Q3 sur un anneau SDH, dans laquelle:

- les protocoles CMISE/CMIP sont pris en charge au niveau de l'interface;
- seuls des sous-réseaux et des connexions de sous-réseaux sont gérés. Ceci peut être déduit aisément des spécifications du contrat d'entreprise;
- seules les fonctions de configuration et de supervision de connexion de sous-réseaux sont envisagées. Ceci peut être déduit aisément des spécifications du contrat d'entreprise;
- seule la vue du client est décrite.

En conséquence, un modèle Ensemble appelé "configuration et gestion de connexion de sous-réseau dans un anneau SDH" peut être décrit comme se constituant de:

- 1) besoins exprimés dans les spécifications de services du contrat d'entreprise;
- 2) scénarios exprimés dans les spécifications de services du contrat d'entreprise (services de base et services étendus);
- 3) ressources déduites des spécifications de services du contrat d'entreprise et des spécifications d'information des services;
- 4) spécifications d'objets gérés qui font partie des spécifications d'ingénierie;
- 5) déclarations de conformité d'objets gérés.

Les directives de modélisation doivent être définies et suivies pour les spécifications d'objets gérés. En particulier, des différences peuvent apparaître entre des modèles en fonction du mode de réalisation du mappage des relations entre objets. Un exemple est donné par le choix de la mise en œuvre de la relation entre le réalisateur d'une connexion et le sous-réseau, soit par corrélation de nom entre les classes d'objets gérés, soit dans des paquetages conditionnels qui peuvent être importés dans des classes d'objets gérés (il convient de noter que l'élément CMISE a été sélectionné pour les directives d'ingénierie, en liaison étroite avec les directives GDMO complètes).

Les définitions des classes d'objets gérés et des paquetages sont les suivantes:

```
subnetwork MANAGED OBJECT CLASS
  DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
  CONDITIONAL PACKAGES
    sncConfigurationPackage PACKAGE
    BEHAVIOUR
      sncConfigurationPackageBehaviour BEHAVIOUR
      DEFINED AS "...";
    PRESENT IF "...";;
REGISTERED AS {...};

subnetworkConnection MANAGED OBJECT CLASS
  DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
  CONDITIONAL PACKAGES
    sncMonitoringPackage PACKAGE
    BEHAVIOUR
      sncMonitoringPackageBehaviour BEHAVIOUR
      DEFINED AS "...";
    PRESENT IF "...";;
REGISTERED AS {...};

sncConfigurationPackage PACKAGE
  BEHAVIOUR ...
  ACTIONS
    setupSNCPPointToPoint,
    setupSNCPPointToMultiPoint,
    releaseSNC;
REGISTERED AS {...};

sncMonitoringPackage PACKAGE
  BEHAVIOUR ...
  NOTIFICATIONS
    operationalStateValueChangeNotification;
REGISTERED AS {...};

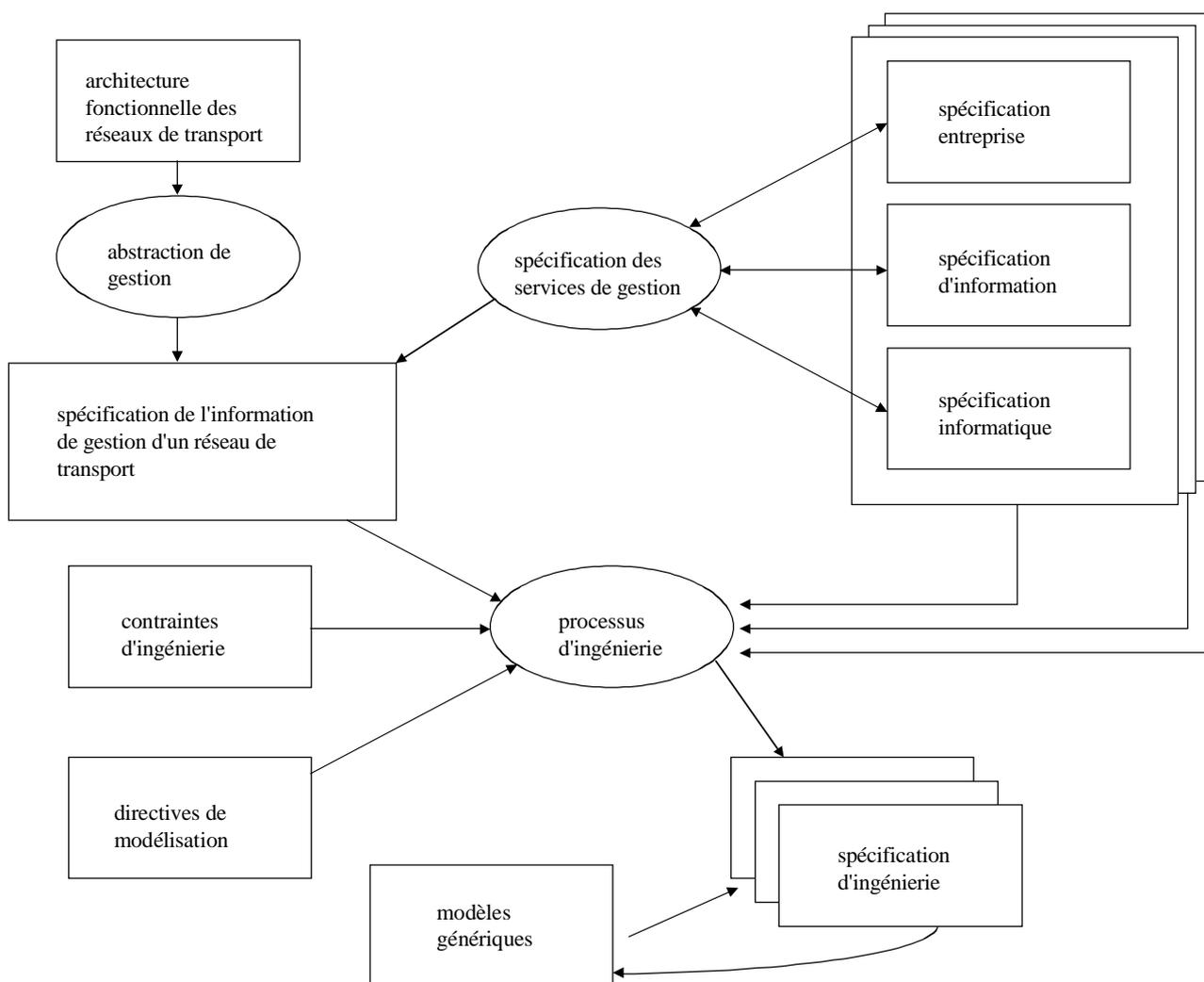
setupSNCPPointToPoint ACTION
  BEHAVIOUR
    setupSNCPPointToPointBehaviour BEHAVIOUR
    DEFINED AS "Voir la définition du comportement de l'opération
      setupSNCPPointToPoint de l'interface informatique semConfiguration";;
  MODE CONFIRMED;
  WITH INFORMATION SYNTAX Voir l'expression INPUT PARAMETER de
    l'opération setupSNCPPointToPoint;
  WITH REPLY SYNTAX Voir l'expression OUTPUT PARAMATERS de l'opération
    setupSNCPPointToPoint;
REGISTERED AS {...};
```

Une convergence doit exister autant que possible entre les spécifications GDMO résultantes et les bibliothèques de classes d'objets gérés, de l'une des deux manières suivantes: par réutilisation des classes d'objets gérés dans des buts particuliers, ou au moyen de bibliothèques de classe d'objet gérés étendues en partant des modèles spécifiques.

APPENDICE V

Exemple de processus d'élaboration de spécification

Ce processus peut être utilisé pour élaborer un ensemble complet de spécifications basées sur la présente méthodologie. Voir la Figure V.1.



T1521180-96

Figure V.1/G.851.1 – Processus d'élaboration de spécification

APPENDICE VI

Mappage entre perspectives

VI.1 Démarche

Le processus de déduction utilisé pour obtenir les spécifications pour le modèle de niveau réseau en utilisant le cadre général du modèle RM-ODP a fourni comme résultat diverses relations entre les éléments (par exemple les rôles, objets d'information ou schémas statiques) associés aux diverses perspectives. Il est intéressant de caractériser ces relations entre perspectives au moyen de deux types principaux de relation: l'*action* et la *référence*. Les relations d'action peuvent être encore subdivisées en celles qui exercent une action forte (ou déterminante) ou une action faible (ou influence). Une relation de référence peut emprunter ou mapper tout ou partie de l'information d'un élément au profit d'un autre. En vue d'éviter une redondance, les relations ne sont décrites que dans le contexte de l'élément situé dans la perspective qui est la plus "proche" de la spécification d'implémentation (ingénierie). Il s'ensuit qu'aucune relation entre perspectives n'est considérée dans le contexte de la perspective entreprise.

La Figure VI.1 décrit les relations du type *référence* et *déterminante* entre perspectives. Une relation déterminante est indiquée par une flèche allant de l'élément influençant vers l'élément influencé. Une relation de référence est indiquée par une ligne sans flèches. Chacune de ces relations ainsi que d'autres relations d'influence sont décrites dans le sous-paragraphe qui suit dans la perspective de l'élément qui est le plus proche de la spécification de l'implémentation. La perspective entreprise n'est pas traitée par ce modèle car elle est traitée dans un autre contexte utilisant le modèle Ensemble.

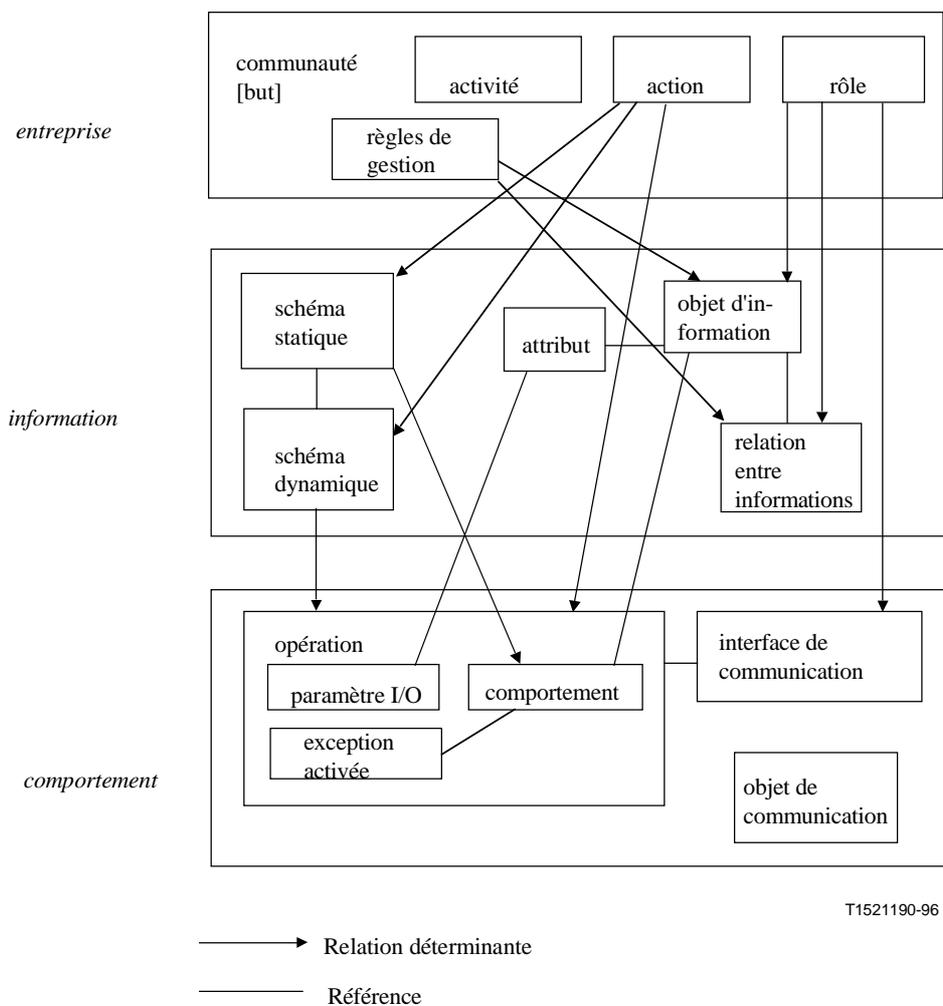


Figure VI.1/G.851.1 – Relation entre perspectives

VI.2 Mappages de la perspective informationnelle

Les éléments clés de la perspective informationnelle sont les objets et attributs d'information, les relations d'information, le schéma statique et le schéma dynamique.

VI.2.1 Objets et relations d'information

Cette sélection d'objets et de relations d'information pour la perspective informationnelle est déterminée avant tout par les rôles de ressource définis dans la perspective entreprise. Toutefois, des objets et relations d'information peuvent également être déterminés par des règles de gestion de communauté, des règles de gestion d'action et des règles de gestion d'activité. Les attributs spécifiques associés aux objets d'information sont déterminés par le but de la communauté et/ou les actions de la communauté. La nécessité d'inclure des attributs dans la perspective informationnelle peut être affectée par les règles de gestion de communauté et les règles de gestion d'action.

VI.2.2 Schéma statique

La définition d'un schéma statique dans la perspective informationnelle est déterminée par une ou plusieurs règles de gestion d'action de communauté entreprise. Les rôles associés à un schéma statique font référence à des objets d'information, les invariants font référence à des relations entre informations. Les règles de gestion de communauté entreprise affectent le schéma statique.

VI.2.3 Schéma dynamique

Les schémas dynamiques sont définis afin de fournir un moyen de faire référence à des transitions particulières entre schémas statiques. Les schémas dynamiques sont déterminés par des actions.

VI.3 Mappages de la perspective informatique

Les éléments clés de la perspective informatique sont les opérations, les interfaces informatiques et les objets informatiques. Les opérations sont caractérisées par des paramètres en entrée, des paramètres en sortie, des exceptions activées et un comportement. Les interfaces informatiques représentent des assemblages d'opérations. Les objets informatiques représentent des assemblages d'interfaces informatiques avec des définitions de contrainte sur les interfaces. Le besoin de permettre une répartition est un moteur clé de la perspective informatique, mais n'est pas inclus dans la description des relations entre perspectives.

VI.3.1 Opération informatique

La nature générale d'une opération ainsi que celle de tout attribut d'opération sont déterminées par l'action de communauté (perspective entreprise) et par le schéma dynamique (perspective informationnelle). Une opération donnée peut représenter une ou plusieurs actions de communauté. Les paramètres d'entrée et de sortie font tous deux référence à des attributs de perspective informationnelle. Les exceptions activées sont en général déterminées par des invariants du schéma statique. Le comportement est déterminé principalement sur la base du schéma statique de la perspective informationnelle. En particulier les préconditions et postconditions d'une clause BEHAVIOUR d'opération sont déterminées par un schéma statique; les clauses PARAMETER MATCHING associent (au moyen de références) des rôles particuliers de chaque schéma statique à des objets d'information particuliers.

VI.3.2 Interface informatique

La définition d'une interface informatique sous l'aspect de fonctions qu'elle comprend est déterminée par des rôles définis dans la perspective entreprise. La nature de l'interface (client ou serveur) est également déterminée par le type de rôle.

VI.3.3 Objet informatique

La définition d'un objet informatique comme collection d'interfaces informatiques est déterminée principalement par les besoins de la perspective informatique, et n'a pas, en conséquence, de relation forte avec des éléments d'autres perspectives.

APPENDICE VII

Directives pour l'utilisation de la notation Z dans la perspective informationnelle

VII.1 Introduction

Les parties formelles de la spécification de la perspective informationnelle ont été spécifiées en utilisant la notation Z. Le schéma de la notation Z n'est pas utilisé directement pour décrire les types d'objets et de relations définissant la perspective informationnelle. La spécification est plutôt basée sur deux ensembles (OBJECT et RELATIONSHIP) qui sont des abstractions de toutes les instances possibles d'objet et de relation dans une application. Le système est décrit sous la forme d'un ensemble de mappages des ensembles de base vers des caractéristiques d'objet et de relation. L'ensemble constitué de ces mappages décrit les objets et relations effectifs qui constituent le système à un instant donné.

Les attributs d'un objet et les entités jouant des rôles dans une relation doivent être déterminés à partir des mappages adéquats. Les schémas Z sont utilisés pour regrouper ces mappages d'une manière homogène avec les concepts de déclaration de type d'objet, comme décrit en détail ci-dessous. Tout type (attribut, objet ou relation) est défini par deux schémas Z. Le premier déclare les composants nécessaires à la définition du type et en formule les invariants statiques, alors que le deuxième définit les transitions d'état autorisées pour le type. Le nom du premier schéma est constitué par convention du nom du type suivi du suffixe **_Static**, alors que le nom du second schéma contient le même nom de type suivi du suffixe **_Dynamic**.

VII.2 Présentation de la notation Z

VII.2.1 Schémas

La notation Z est une notation formelle basée sur la théorie des ensembles et la logique de prédicats du premier ordre. Le concept de modélisation de base de la notation Z est l'ensemble, à partir duquel peuvent être construites des structures plus complexes. Une fonction, par exemple, est un ensemble (éventuellement non fini) de couples ordonnés. Un ensemble peut être un ensemble déterminé (sans que la structure interne de ses membres soit visible), peut être défini par extension (en énumérant ses éléments) ou peut être défini par son contenu (en fournissant un ensemble de base et un prédicat auquel doivent satisfaire tous les éléments potentiels). L'encapsulation est fournie dans la notation Z par la notation de schéma pouvant être utilisée d'un certain nombre de manières. Dans l'état actuel de la spécification, les schémas ne sont utilisés que pour encapsuler des déclarations et des invariants à des fins d'étiquetage et de réutilisation. Un schéma nommé de la notation Z se présente sous la forme suivante:

_____ Schema-name _____
Declaration
Predicate

dans laquelle

- déclaration est composée d'une liste de variables et de leurs types (les ensembles auxquels appartiennent leurs valeurs);
- predicate est une liste de conditions, éventuellement vide, auxquelles les valeurs de la variable doivent satisfaire.

Après cette déclaration, le schéma **schema-name** peut être utilisé dans la partie déclaration d'un schéma ultérieur, ce qui a pour effet d'ajouter les entités **declaration** et **predicate** aux parties déclaration et prédicat du schéma pour lequel est fait l'inclusion.

Une convention Z normalisée, utilisée en général pour décrire des transitions d'état, consiste à déclarer pour Δ **schema-name** deux copies des variables contenues dans le schéma **schema-name**, l'un des ensembles de variables étant affectée d'un signe prime ($\hat{\text{O}}$). Les variables sans prime sont interprétées avant un état avant, celles avec prime comme un état après.

VII.2.2 Symboles

La notation Z fournit une grande variété de notations pour les ensembles et les fonctions logiques, ce qui permet une spécification concise des données et des aspects fonctionnels d'un système. La spécification des objets d'information et des relations n'utilise qu'une partie restreinte de ces symboles, résumée ci-dessous:

- 1) \in : appartenance à un ensemble.

- 2) \cup : union d'ensembles.
- 3) \cap : intersection d'ensembles.
- 4) \subseteq : relation d'inclusion.
- 5) $\#$: nombre d'éléments d'un ensemble.
- 6) $\dashv\vdash$: fonction partielle. Si X et Y sont des ensembles, $X \dashv\vdash Y$ est l'ensemble des fonctions partielles de X vers Y , qui sont les relations liant chaque élément x de X vers un élément au plus de Y .
- 7) **dom, ran**: domaine et étendue d'une relation (une fonction, en particulier). Si R est une relation binaire entre X et Y , le domaine de R (**dom** R) est l'ensemble de tous les éléments de X qui sont liés avec au moins un élément de Y par la relation R . L'étendue (des valeurs) de R (**ran** R) est l'ensemble de tous les éléments de Y auxquels au moins un élément de X est lié par la relation R .
- 8) \sim : inverse d'une relation. Si la relation R fait correspondre a à b , alors $R\sim$ fait correspondre b à a .
- 9) $(| \)$: image d'une relation. Si R est une relation de X vers Y , alors pour tout sous-ensemble S de X , $R(|S)$ est l'ensemble de valeurs dans Y qui sont liées par la relation R à une valeur appartenant à S .
- 10) \vee : disjonction logique (ou).
- 11) \wedge : conjonction logique (et).
- 12) \Leftrightarrow : implication logique.
- 13) \forall : quantificateur universel. Le prédicat $\forall x : S \mid \text{pre} \bullet \text{cond}$, dans lequel pre et cond sont des expressions logiques, s'énonce "pour tout élément x de l'ensemble S satisfaisant à la condition pre , le prédicat cond est vrai"
- 14) \exists : quantificateur existentiel. Le prédicat $\exists x : S \mid \text{pre} \bullet \text{cond}$ s'énonce "il existe un élément x de l'ensemble S satisfaisant à la condition pre , pour lequel le prédicat cond est vrai"
- 15) **F**: puissance finie d'un ensemble. Si S est un ensemble, **F** S est l'ensemble de tous les sous-ensembles de S .
- 16) Δ : delta ("opération") convention de dénomination de schéma (voir VII.2.1)

NOTE – Dans le présent appendice, beaucoup de symboles ont un aspect légèrement différent de celui des spécifications réelles dans lesquelles des jeux de caractères spéciaux ont été utilisés.

VII.2.3 Exemple

Le fragment de notation Z ci-dessous définit deux ensembles donnés X et Y , puis déclare un schéma S avec deux variables: a , de type X (c'est-à-dire élément de X), et f , une fonction partielle de X dans Y . Le prédicat de S énonce que a est un des éléments de X mappés par f , et qu'il existe au moins deux éléments de Y qui sont des cibles du mappage de la fonction f .

[X, Y]

S
$a: X$ $f: X \dashv\vdash Y$
$a \in \text{dom } f \wedge \#(\text{ran } f) > 1$

VII.3 Conventions de spécification

La spécification d'un système se constitue des attributs que peuvent posséder les objets, des classes d'objets et des classes de relations. La notation de schéma Z est utilisée pour fournir des déclarations encapsulées de tous les mappages nécessaires pour spécifier un type donné. Ces mappages ont des domaines au sein des ensembles **OBJECT** (l'ensemble de tous les objets potentiels) et **RELATIONSHIP** (ensemble de toutes les relations potentielles).

[OBJECT, RELATIONSHIP]

VII.3.1 Spécification d'attribut

Un type d'attribut est modélisé par une fonction partielle (portant le même nom que le type) sur l'ensemble des objets potentiels. Le domaine de la fonction est l'ensemble des instances de toute classe d'objets possédant cet attribut. Les valeurs que la fonction peut prendre sont les valeurs que peuvent prendre les attributs. Cette étendue de valeurs, déclarée suffixé par le schéma statique, peut être spécifiée par les opérateurs usuels d'énumération d'ensemble et d'expression de la notation Z.

Un type d'attribut peut faire l'objet de contraintes de comportement dynamique. Celles-ci sont déclarées dans le schéma dynamique qui fait référence aux états avant et après.

A titre d'exemple, l'attribut **number** peut être déclaré comme suit:

_____ number_Static _____
number: OBJECT --+--> Integer
∇ n: ran number • n > 5

Si l'attribut **number** d'un objet n'est jamais modifié, le schéma dynamique peut être le suivant:

_____ number_Dynamic _____
Δnumber_Static
∇ obj: dom number ∪ dom number' • number'(obj) = number(obj)

indiquant qu'un objet qui existe avant et après une transition d'état du système ne peut modifier son attribut **number**.

VII.3.2 Spécification d'objet

Un schéma décrivant les attributs d'une instance unique d'un objet n'est pas défini dans le style de spécification utilisé dans ce document. En remplacement, un ensemble d'objets (de même nom que le type) est défini pour chaque type d'objet. Cette classe d'objets est l'ensemble de tous les objets existant dans le système qui satisfont aux propriétés du type d'objet (dans la terminologie ODP, une classe d'objets est l'ensemble de toutes les instances existantes d'un type d'objet). La notion d'instanciation (au sens ODP) n'est pas modélisée dans la perspective informationnelle. En outre, un schéma Z qui définit une classe d'objets n'est pas un squelette qui permet l'instanciation d'un objet appartenant à cette classe.

Il existe plusieurs conventions de structuration qui doivent être appliquées pour décrire les attributs d'objets, l'héritage et les invariants.

attributs

Il est nécessaire, pour spécifier que tous les objets d'une classe possèdent un attribut donné, d'inclure le schéma `_Static` de cet attribut dans la clause de déclaration du schéma `_Static` de l'objet (d'une manière explicite ou par inclusion d'un schéma d'une autre classe d'objets). Le prédicat du schéma de l'objet doit en outre assurer que le mappage de l'attribut s'applique à tout membre de la classe (c'est-à-dire que les membres de la classe constituent un sous-ensemble du domaine de mappage).

sous-types

La spécification d'une relation de sous-type (représentée dans la directive `GDMO` par le mécanisme d'héritage entre classes) nécessite que le schéma statique de toute superclasse (plus d'une en cas d'héritage multiple) soit inclus dans la clause de déclaration du schéma statique de l'objet. Le prédicat du schéma de l'objet doit en outre assurer que les membres de la sous-classe appartiennent également tous à la superclasse ou aux superclasses.

invariants

Les contraintes concernant les combinaisons autorisées des valeurs d'attributs d'un objet peuvent être exprimées dans le prédicat du schéma statique de l'objet.

transitions

La spécification des transitions d'état valides pour un type d'objet nécessite la spécification d'un schéma dynamique pour ce type. La signature du schéma doit être le schéma `delta Z` habituel (décrivant les copies avec et sans primes des variables statiques), complété comme suit: pour tout schéma inclus dans le schéma statique afin de décrire l'héritage et les attributs du type, un schéma dynamique correspondant doit être inclus dans le schéma dynamique de l'objet. D'autres contraintes peuvent en outre être ajoutées au prédicat du schéma dynamique de l'objet.

exemple

Soit un type d'objet **super** déjà défini. Une sous-classe **sub** qui ajoute un attribut supplémentaire pourrait être définie de la manière suivante:

_____ sub_Static _____
sub: F OBJECT super_Static number_Static
sub subseteq super sub subseteq dom number
_____ sub_Dynamic _____
Δ sub_Static super_Dynamic number_Dynamic

VII.3.3 Spécification de relation

De même, un type de relation est spécifié par la description de la classe de relation sous la forme d'un ensemble d'instances (dans ce cas, il s'agit d'un sous-ensemble de l'ensemble **RELATIONSHIP** et non de l'ensemble **OBJECT**), ainsi que par les mappages qui, dans ce cas, décrivent des rôles aussi bien que des attributs.

rôles

Pour spécifier que toutes les relations d'une classe impliquent un rôle particulier, le schéma statique de la relation doit déclarer (directement ou par héritage) une fonction partielle de nom approprié portant sur l'ensemble de toutes les instances possibles de relation. Le prédicat du schéma de la relation doit en outre assurer que le mappage du rôle s'applique à tout membre de la classe de relations.

compatibilité de rôle

Les classes d'objets compatibles avec un rôle sont spécifiées par la présence d'un ensemble approprié de conditions d'inclusion dans le schéma statique du prédicat de la relation. Pour un type donné de relation, le domaine de la fonction partielle modélisant un rôle se limite à un sous-ensemble de l'union des classes d'objet compatibles avec ce rôle.

cardinalité de rôle

La cardinalité d'un rôle peut être spécifiée de deux manières. Dans le premier cas, la déclaration du rôle peut impliquer une certaine cardinalité: si la fonction modélisant le rôle a comme domaine l'ensemble OBJECT, le rôle possède une cardinalité implicite égale à (1..1), si la fonction a comme domaine l'ensemble **F** OBJECT (l'ensemble des sous-ensembles OBJET), le rôle possède une cardinalité implicite égale à (0..N). Dans ce dernier cas, un prédicat peut imposer une limitation supplémentaire à la cardinalité.

sous-type

Les sous-types sont représentés, comme les types d'objets, par l'inclusion dans le schéma d'un ou de plusieurs supertypes et prédicats assurant que la nouvelle classe de relations est un sous-ensemble de la superclasse ou des superclasses. Les types d'objets autorisés à jouer un rôle hérité peuvent en outre être limités à un sous-ensemble des types d'objets hérités par l'addition d'un prédicat portant sur le domaine du mappage de rôle adéquat.

invariants et attributs

Les invariants d'une relation sont spécifiés dans le prédicat du schéma statique. Etant donné que ces invariants impliquent généralement les attributs des objets jouant les rôles, le schéma statique de tout attribut auquel fait référence un invariant doit être inclus dans les déclarations du schéma statique de relation.

transitions

Comme dans le cas des transitions d'objet, le schéma dynamique d'une relation est basé sur un delta de son schéma statique, et il doit exister dans le schéma dynamique de la relation une déclaration dynamique correspondant à tout schéma inclus dans le schéma statique.

exemple

Soit une relation impliquant deux rôles: **firstRole** joué par un unique objet de type **super** et un rôle joué par au moins deux objets de type **secondRole**, chacun d'eux ayant une valeur 10 comme attribut **number**. Ceci peut être spécifié de la manière suivante:

rel_Static

rel: F RELATIONSHIP
firstRole: RELATIONSHIP --+--> OBJECT
secondRole: RELATIONSHIP ---+--> F OBJECT
super_Static
sub_Static
number_Static

rel subseteq **dom** firstRole
rel subseteq **dom** secondRole

$\forall R: \text{rel} \bullet$
firstRole(R) \in super \wedge
secondRole(R) \in F sub

$\forall R: \text{rel} \bullet$
#(secondRole R) > 1

$\forall R: \text{rel} \bullet$
 $\forall s: (\text{secondRole R}) \bullet \text{number}(s) = 10$

(Il convient de noter qu'il existe des façons plus concises de spécifier les propriétés indiquées ci-dessus, il est cependant important d'utiliser une structure systématique permettant d'identifier aisément les invariants décrits dans la spécification non formalisée.)

rel_Dynamic

Δ rel_Static
super_Dynamic
sub_Dynamic
number_Dynamic

SERIES DES RECOMMANDATIONS UIT-T

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