



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**M.3020**

(02/2000)

SÉRIE M: RGT ET MAINTENANCE DES RÉSEAUX:  
SYSTÈMES DE TRANSMISSION, DE TÉLÉGRAPHIE,  
DE TÉLÉCOPIE, CIRCUITS TÉLÉPHONIQUES ET  
CIRCUITS LOUÉS INTERNATIONAUX

Réseau de gestion des télécommunications

---

**Méthodologie pour la spécification des  
interfaces du réseau de gestion des  
télécommunications**

Recommandation UIT-T M.3020

(Antérieurement Recommandation du CCITT)

---

RECOMMANDATIONS UIT-T DE LA SÉRIE M

**RGT ET MAINTENANCE DES RÉSEAUX: SYSTÈMES DE TRANSMISSION, DE TÉLÉGRAPHIE, DE  
TÉLÉCOPIE, CIRCUITS TÉLÉPHONIQUES ET CIRCUITS LOUÉS INTERNATIONAUX**

Introduction et principes généraux de maintenance et organisation de la maintenance	M.10–M.299
Systèmes de transmission internationaux	M.300–M.559
Circuits téléphoniques internationaux	M.560–M.759
Systèmes de signalisation à canal sémaphore	M.760–M.799
Systèmes internationaux de télégraphie et de phototélégraphie	M.800–M.899
Liaisons internationales louées par groupes primaires et secondaires	M.900–M.999
Circuits internationaux loués	M.1000–M.1099
Systèmes et services de télécommunication mobile	M.1100–M.1199
Réseau téléphonique public international	M.1200–M.1299
Systèmes internationaux de transmission de données	M.1300–M.1399
Appellations et échange d'informations	M.1400–M.1999
Réseau de transport international	M.2000–M.2999
<b>Réseau de gestion des télécommunications</b>	<b>M.3000–M.3599</b>
Réseaux numériques à intégration de services	M.3600–M.3999
Systèmes de signalisation par canal sémaphore	M.4000–M.4999

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

**Méthodologie pour la spécification des interfaces du réseau  
de gestion des télécommunications**

**Résumé**

La présente Recommandation UIT-T décrit la méthode d'analyse et conception des besoins unifiés du RGT (UTRAD, *unified TMN requirements, analysis and design*) de spécification des interfaces du RGT. Elle décrit le processus permettant de déterminer les spécifications d'interfaces à partir des besoins, de l'analyse et de la conception (RAD, *requirements, analysis and design*). Des lignes directrices sont données en vue de décrire les phases RAD en langage de modélisation unifié (UML, *unified modelling language*); toutefois, elles n'interdisent pas d'autres techniques de spécification d'interface. Les lignes directrices relatives à l'utilisation du langage UML sont décrites à haut niveau dans la présente Recommandation UIT-T.

**Source**

La Recommandation M.3020 de l'UIT-T, élaborée par la Commission d'études 4 (1997-2000) de l'UIT-T, a été approuvée le 4 février 2000 selon la procédure définie dans la Résolution 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 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

## NOTE

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

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2001

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

		<b>Page</b>
1	Introduction.....	1
1.1	Domaine d'application .....	1
1.2	Recommandations connexes.....	1
1.3	Abréviations.....	2
1.4	Définitions .....	2
	1.4.5 Termes importés de la Recommandation UIT-T M.3010.....	3
	1.4.6 Termes importés du langage UML .....	3
1.5	Fondements de la méthodologie et de la notation.....	3
1.6	Utilisation de la notation UML.....	4
2	Méthodologie .....	4
2.1	Généralités .....	4
2.2	Application et structure de la méthodologie .....	4
2.3	Méthodologie détaillée.....	5
	2.3.1 Besoins.....	5
	2.3.2 Analyse .....	6
	2.3.3 Conception.....	6
2.4	Spécifications d'interface RGT .....	7
2.5	Traçabilité dans le processus UTRAD.....	7
2.6	Structure de la documentation .....	7
	Annexe A – Directives pour la définition de l'interface de gestion (GDMI) .....	8
A.1	Introduction.....	8
A.2	Gabarit GDMI.....	8
	A.2.1 Domaine d'application .....	8
	A.2.2 Spécifications.....	8
	Annexe B – Règles d'affectation des identificateurs d'objet RGT .....	10
B.1	Structure des identificateurs d'objet RGT.....	10
B.2	Extension de la structure des identificateurs d'objet RGT aux "parties" de Recommandation .....	12
B.3	Procédures d'affectation appliquées au RGT .....	12
B.4	Affectation des identificateurs d'objet dans un contexte d'application RGT .....	13
	Annexe C – Opérations génériques.....	14
C.1	Opérations génériques.....	14
	C.1.1 getAttributes .....	14
	C.1.2 getAllAttributes .....	14
	C.1.3 notifications .....	14

	<b>Page</b>
Appendice I – Exemple d'utilisation des directives GDMI (fourniture de LCS) .....	16
I.1 Introduction.....	16
I.2 Gabarit GDMI.....	16
I.2.1 Domaine d'application .....	16
I.2.2 Besoins.....	16
I.2.3 Conception.....	26

## Recommandation UIT-T M.3020

### Méthodologie pour la spécification des interfaces du réseau de gestion des télécommunications

## 1 Introduction

### 1.1 Domaine d'application

La présente Recommandation UIT-T décrit la méthode d'analyse et conception des besoins unifiés du RGT (UTRAD, *unified TMN requirements, analysis and design*) de spécification des interfaces du RGT. Elle décrit le processus permettant de déterminer les spécifications d'interfaces à partir des besoins, de l'analyse et de la conception (RAD, *requirements, analysis and design*). Des lignes directrices sont données en vue de décrire les phases RAD en langage de modélisation unifié (UML, *unified modelling language*); toutefois elles n'interdisent pas les autres techniques de spécification d'interface. Les lignes directrices relatives à l'utilisation du langage UML sont décrites à haut niveau dans la présente Recommandation UIT-T. D'autres Recommandations UIT-T de la présente série contiendront une définition plus détaillée concernant l'utilisation spécifique de la notation UML dans le RGT.

Une spécification d'interface s'applique au(x) service(s) de gestion défini(s) dans la Recommandation UIT-T M.3200. Une telle spécification peut prendre en charge une partie d'un ou de plusieurs services de gestion. Les services de gestion comprennent des fonctions de gestion. Ces fonctions peuvent renvoyer à celles définies dans la Recommandation UIT-T M.3400 ou se spécialiser en vue de s'adapter à un domaine de gestion spécifique ou bien de nouvelles fonctions peuvent être identifiées au besoin.

### 1.2 Recommandations connexes

Il convient de se référer aux Recommandations connexes suivantes:

- [1] Recommandation UIT-T M.3010 (2000), *Principes des réseaux de gestion des télécommunications*.
- [2] Recommandation UIT-T M.3200 (1997), *Services de gestion du RGT et domaines gérés des télécommunications: aperçu général*.
- [3] Recommandation UIT-T M.3400 (2000), *Fonctions de gestion RGT*.
- [4] Langage de modélisation unifié, Section 1 de *OMG Modelling*, OMG Doc. No. Formal/99-06-01.
- [5] Recommandation UIT-T M.3208.1 (1997), *Services de gestion RGT pour réseaux à circuits spécialisés et circuits reconfigurables: services de circuits loués*.
- [6] Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.
- [7] Recommandation UIT-T Z.100 (1999), *SDL: Langage de description et de spécification*.

### 1.3 Abréviations

La présente Recommandation utilise les abréviations suivantes.

ASN.1	notation de syntaxe abstraite numéro un ( <i>abstract syntax notation one</i> )
CMIP	protocole commun d'informations de gestion ( <i>common management information protocol</i> )
CNM	gestion de réseau client ( <i>customer network management</i> )
CORBA	architecture du courtier de requêtes pour objets communs ( <i>common object request broker architecture</i> )
GDMI	directives pour la définition des interfaces de gestion ( <i>guidelines for the definition of management interface</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> )
LCS	service de circuits loués ( <i>leased circuit service</i> )
NE	élément de réseau ( <i>network element</i> )
OAM&P	exploitation, administration, maintenance et fourniture ( <i>operations, administration, maintenance and provisioning</i> )
OMG	groupe de gestion d'objets ( <i>object management group</i> )
OO	orienté(e) objet
OS	système d'exploitation ( <i>operations system</i> )
OSI	interconnexion des systèmes ouverts ( <i>open systems interconnection</i> )
RGT	réseau de gestion des télécommunications
SC	client du service ( <i>service customer</i> )
SDL	langage de description et de spécification ( <i>specification and description language</i> )
SLA	accord de niveau de service ( <i>service level agreement</i> )
SP	fournisseur du service ( <i>service provider</i> )
UML	langage de modélisation unifié ( <i>unified modelling language</i> )
UTRAD	analyse et conception des besoins unifiés du RGT ( <i>unified TMN requirements, analysis and design</i> )

### 1.4 Définitions

La présente Recommandation définit les termes suivants:

**1.4.1 objectifs de gestion du RGT:** objectifs de haut niveau d'un utilisateur exécutant des activités de gestion.

**1.4.2 rôles de gestion du RGT:** définissent les activités attendues du personnel ou d'un système qui réalise la gestion de télécommunication. Les rôles de gestion RGT sont définis indépendamment des autres composantes, c'est-à-dire des ressources de télécommunication et des fonctions de gestion du RGT.

**1.4.3 ressources de télécommunication:** entités physiques ou logiques nécessitant une gestion qui utilisent des services de gestion RGT.

**1.4.4 scénario de gestion RGT:** exemple d'interactions de gestion de la part d'un service de gestion.

#### **1.4.5 Termes importés de la Recommandation UIT-T M.3010**

Les termes suivants de la Recommandation UIT-T M.3010 [1] sont utilisés dans la présente Recommandation UIT-T.

- utilisateur;
- service de gestion RGT;
- ensemble de fonctions de gestion RGT.

#### **1.4.6 Termes importés du langage UML**

Les termes UML [4] suivants sont utilisés dans la présente Recommandation UIT-T.

- diagramme d'activité;
- acteur;
- classe;
- diagramme de classe;
- diagramme de collaboration;
- diagramme séquentiel;
- diagramme d'état;
- stéréotype;
- cas d'utilisation.

### **1.5 Fondements de la méthodologie et de la notation**

Les objectifs qui ont servi à l'élaboration de la méthodologie et au choix d'une notation sont les suivants.

- 1) La notation et la méthodologie doivent prendre en compte tous les paramètres de l'espace problème, à savoir la gestion des télécommunications.
- 2) La notation doit faciliter la production non ambiguë de la spécification dans les paradigmes cibles de gestion de réseau spécifiés dans la Recommandation UIT-T Q.812.
- 3) Des points de conformité non optionnels doivent être spécifiés dans les trois phases. S'il est nécessaire d'avoir des fonctionnalités optionnelles pour prendre en charge l'espace problème de télécommunication, ces fonctionnalités seront spécifiées. Pour les phases besoins et analyse, les fonctionnalités optionnelles autorisées seront spécifiées dans cette série de Recommandations UIT-T. Les fonctionnalités optionnelles admissibles pour la phase de conception seront spécifiées dans les Recommandations UIT-T de la série Q.81x.
- 4) Il doit être possible de produire, à partir des spécifications de conception, des définitions interopérables propres au langage (par exemple l'UML avec l'IDL, l'UML avec le GDMO/ASN.1).

La notation courante choisie, comme indiqué plus loin ne correspond pas à tous les objectifs précités. Cependant, on s'attend à ce que ces objectifs soient remplis étant donné que ces notations sont déjà largement utilisées dans l'industrie.

Les fonctionnalités optionnelles qui doivent être prises en charge incluent les fonctionnalités de la notation pour les trois phases considérées et pour les capacités propres aux paradigmes cibles de gestion des réseaux (par exemple la sélection des fonctionnalités CORBA et les fonctionnalités

optionnelles dans une fonctionnalité donnée). Ces caractéristiques ne sont pas incluses dans la méthodologie mais devraient être incluses dans d'autres Recommandations UIT-T.

## 1.6 Utilisation de la notation UML

Le Tableau 1 donne la correspondance entre les concepts du RGT et la notation UML. La présente Recommandation UIT-T spécifie les concepts et les notations de haut niveau à utiliser dans les différentes phases. D'autres Recommandations de la présente série décriront les directives pour utiliser des aspects spécifiques des notations, des extensions requises ainsi que les stéréotypes adaptés à l'utilisation au sein du RGT.

Des stéréotypes sont utilisés pour étendre la notation UML. Les stéréotypes approuvés à utiliser dans l'environnement RGT sont inclus dans la présente Recommandation UIT-T (voir Annexe C).

**Tableau 1/M.3020 – Concepts associés aux besoins**

Concept RGT	Notation UML	Commentaire
utilisateur	acteur	Un utilisateur est modélisé comme un acteur.
rôle de gestion	acteur	Un acteur joue un rôle. Il est conseillé de modéliser un seul rôle par acteur.
fonction de gestion	cas d'utilisation	Une fonction de gestion est modélisée par un ou plusieurs cas d'utilisation.
ensemble de fonctions de gestion	cas d'utilisation	Un ensemble de fonctions de gestion est un cas d'utilisation composite dont chaque fonction de gestion est (potentiellement) modélisée par un cas d'utilisation distinct.
service de gestion	cas d'utilisation	Un service de gestion est modélisé par un cas d'utilisation de haut niveau.
scénario de gestion	diagramme de séquence	Les diagrammes de séquence sont préférés aux diagrammes de collaboration.
type de ressource de télécommunication	classe	Les diagrammes de classe décrivent les détails des propriétés du type de ressource de télécommunication, à un niveau de détail approprié à la phase de la méthodologie.
objectifs de gestion	–	Les objectifs de gestion sont transcrits sous forme de descriptions textuelles étant donné qu'il n'y a pas de notation UML applicable.

## 2 Méthodologie

### 2.1 Généralités

L'objet de la présente méthodologie est de donner une description des processus conduisant à la définition des interfaces du RGT.

### 2.2 Application et structure de la méthodologie

La méthodologie UTRAD spécifie un processus itératif à trois phases dont les caractéristiques permettent la traçabilité à travers ces trois phases. Les trois phases s'appliquent aux techniques acceptées par l'industrie, utilisant les principes d'analyse et de conception orientés objet. Ces trois phases sont le cahier des charges, l'analyse et la conception. Les techniques doivent permettre

l'utilisation ou le développement d'outils d'aide disponibles sur le marché. Différentes techniques peuvent être utilisées pour les phases selon la nature du problème.

## **2.3 Méthodologie détaillée**

Les phases besoins et analyse produisent les spécifications UML. La phase conception utilise la notation spécifique au paradigme de gestion du réseau. Les résultats des trois phases sont les suivants:

- Phase besoins – Besoins.
- Phase analyse – Spécification indépendante de l'implémentation.
- Phase conception – Spécification spécifique à la technologie.

Initialement, pour la phase conception on utilisera une approche manuelle ou personnalisée. Lorsque les définitions interopérables propres au protocole peuvent être générées au moyen d'outils, les notations UML pourront être appliquées à la phase de conception. Cependant, certaines définitions propres au protocole, telle la hiérarchie de classe, peuvent être décrites au moyen de la notation UML.

Les sous-paragraphes ci-dessous décrivent les trois phases.

### **2.3.1 Besoins**

Les besoins correspondant au problème à résoudre sont divisés en deux classes. La première classe de besoins est appelée ici besoins professionnels. L'expert sur le sujet doit être en mesure de déterminer que les besoins couvrent bien les besoins du problème de gestion à résoudre. La deuxième classe est appelée besoins de la spécification. Ces besoins doivent contenir suffisamment de détails pour permettre que la définition de l'interface dans les phases d'analyse et de conception puisse être élaborée. Etant donné que les définitions finales de l'interface doivent être traçables relativement aux besoins, il peut être nécessaire d'avoir un processus itératif parmi les trois phases. Toute ambiguïté dans les besoins devra être levée par ce processus itératif afin de garantir qu'une spécification implémentable puisse être élaborée.

Différentes techniques peuvent être utilisées pour spécifier les deux classes de besoins. Indépendamment de la technique utilisée, la lisibilité des besoins revêt un caractère essentiel. Il n'est pas exigé que les besoins soient exprimés en notation assimilable par la machine étant donné que la lisibilité et la traçabilité sont possibles. Une approche possible permettant de délimiter les différents besoins pour assurer leur traçabilité consiste à les énumérer.

La phase d'expression des besoins inclut l'identification d'aspects (par exemple, la politique de sécurité, l'étendue du domaine problèmes en termes d'applications, de ressources et de rôles assumés par les ressources). L'Appendice I contient un exemple de besoins. Les besoins définissent les rôles, les responsabilités et les relations entre les entités constituantes pour l'espace problème. Différentes techniques incluant la représentation textuelle peuvent être utilisées pour préciser les besoins professionnels. Afin de faciliter la traçabilité de ces besoins relativement en phase de conception et d'implémentation, il est recommandé d'énumérer les besoins.

Le problème doit être limité par un domaine d'application spécifique. Une façon de déterminer ce domaine d'application est d'utiliser les services de gestion identifiés dans la Recommandation UIT-T M.3200 et les ensembles de fonctions définis dans la Recommandation UIT-T M.3400. Les besoins sont exprimés au moyen des ressources à gérer et des fonctions de gestion. Une amélioration de la Recommandation UIT-T M.3400 peut être nécessaire afin de répondre aux besoins professionnels du problème.

Des cas d'utilisation de l'UML et des scénarios doivent être utilisés pour interagir avec les experts pour comprendre les besoins professionnels. Les besoins doivent également identifier les conditions d'échec visibles au niveau du processus professionnel.

Les besoins énoncés doivent être complets et détaillés. La nature récursive de la méthodologie UTRAD est utilisée pour obtenir cette exhaustivité. Les phases d'analyse et de conception dépendent de la bonne formulation des besoins (clairs et bien documentés).

### **2.3.2 Analyse**

Dans la phase d'analyse, les besoins sont utilisés pour identifier les entités interagissantes, leurs propriétés et leurs interrelations. Ceci permet de définir les interfaces offertes par ces entités. Dans la notation UML, ces entités deviennent des classes. Les descriptions des classes ainsi que les interfaces exposées doivent être traçables relativement aux besoins. La relation entre les classes définies dans la spécification d'analyse et les classes dans la spécification de conception ne sont pas nécessairement biunivoques.

La présente Recommandation UIT-T contient les directives de haut niveau sur la façon d'utiliser la notation UML afin de prendre en charge la spécification d'interface du RGT; cependant, le langage SDL [7], technique bien acceptée par l'industrie, doit être utilisé pour compléter les définitions UML.

La phase d'analyse doit être indépendante des contraintes de conception. Par exemple, l'analyse peut être documentée en utilisant les principes orientés objet même si la conception peut utiliser une technologie non orientée objet. L'information spécifiée dans la phase d'analyse inclut les descriptions de classe, les définitions de données, les relations de classe, les diagrammes d'interaction (les diagrammes séquentiels et/ou les diagrammes de collaboration), les diagrammes de transition d'état et les diagrammes d'activité. Les définitions de classe incluent la spécification des opérations, les signaux (stimulus asynchrones tels la réception d'opérations, événements et exceptions), les attributs et les comportements précisés sous forme de notes ou de descriptions textuelles.

Les définitions génériques (opérations et stéréotypes) contenues dans l'Annexe C sont fournies pour utilisation pendant la phase d'analyse. Elles définissent l'extraction et la définition des attributs multiples et l'émission des notifications. Elles peuvent être incluses dans les définitions de classe et les diagrammes d'interaction.

### **2.3.3 Conception**

Dans la phase de conception, la spécification d'interface interopérable et pouvant être implémentée est produite. Cela impliquerait le choix dû à un langage de spécification cible. Les besoins de la phase de conception dépendent du paradigme spécifique de gestion de réseau RGT.

La sélection de ce paradigme est traitée dans d'autres Recommandations sur le RGT.

Dans le contexte du paradigme du RGT fondé sur la gestion des systèmes OSI, la spécification de conception est la spécification du modèle informationnel au moyen des modèles GDMO des classes d'objets gérés, d'attributs, de comportement, de notifications, d'actions, d'instances de nommage de la classe et des spécifications d'erreur/exception. La syntaxe de l'information est spécifiée au moyen de la notation ASN.1.

En GDMO, la hiérarchie de classe d'objets spécifie les propriétés de classe d'objets qui sont nécessaires pour la gestion. L'utilisation généralisée de l'héritage (super-classe et sous-classe) est nécessaire pour tirer le plus parti de la réutilisation des spécifications. Les classes d'objets sont spécifiées au moyen des modèles définis dans la Recommandation UIT-T X.722, "Structure de l'information de gestion – Directives pour la définition des objets gérés". Les modèles définissant le modèle d'information doivent être enregistrés (conformément aux règles définies dans la Recommandation UIT-T X.722) avec une valeur pour l'identificateur d'objet ASN.1. L'Annexe B décrit la procédure d'assignation des valeurs d'enregistrement. Pour ces classes d'objets qui sont déjà spécifiées dans d'autres Recommandations UIT-T et normes ISO, seule une référence à une Recommandation particulière ou à une classe d'objets particulière est nécessaire. Le nommage ne fait pas partie, ou n'a pas pour objet la hiérarchie de classe d'objets.

Dans le contexte du RGT fondé sur le CORBA, le modèle d'information est défini au moyen du langage IDL.

A mesure que d'autres paradigmes seront ajoutés au RGT les notations/langages définis par ces paradigmes seront utilisés.

Dans la phase de conception, il est recommandé d'accompagner les descriptions UML à partir des phases besoins et analyse de renvois afin d'élargir la spécification comportementale. Par exemple, la définition comportementale du GDMO peut renvoyer à d'autres diagrammes d'états, diagrammes séquentiels et définition de classe dans la phase d'analyse. Au besoin, on peut inclure des diagrammes UML décrivant les interactions entre entités, correspondant à des paradigmes protocolaires spécifiques.

## **2.4 Spécifications d'interface RGT**

Une spécification d'interface RGT inclut les spécifications besoins, analyse et conception décrites en 2.3. Une structure permettant de définir ces spécifications est donnée dans l'Annexe A et est appelée Directives pour la définition de l'interface de gestion (GDMI).

Ces techniques et les notations qui les accompagnent sont également utilisables pour la conception d'un système, aux spécifications d'interface du RGT, même si la conception du système n'est pas considérée comme faisant partie des Recommandations sur le RGT. Elles doivent aider à la description de la façon dont les spécifications d'interface sont appliquées dans la gestion des ressources à l'intérieur d'un système, tel un élément de réseau.

## **2.5 Traçabilité dans le processus UTRAD**

Afin d'obtenir la traçabilité entre besoins, analyse et conception, il est nécessaire de prévoir une identification et des pointeurs appropriés dans chaque élément du modèle. Par exemple, les besoins peuvent être identifiés par des numéros ou des références aux fonctions figurant dans la liste des fonctions de la Recommandation UIT-T M.3400. Il est recommandé de numéroter les autres besoins (nouvelles fonctions non incluses dans la Recommandation UIT-T M.3400) ou la politique de sécurité et les exigences de qualité de fonctionnement, car une spécification de conception peut répondre à ces exigences différemment selon le protocole sur lequel elle repose. Une autre approche utilisée dans l'exemple donné dans l'Appendice I (I.2.2.3) consiste à référencer les cas d'utilisation et les descriptions textuelles associées. Les résultats de la phase d'analyse spécifient, pour les différents cas d'utilisation, les autres besoins en informations détaillées. La phase de conception doit pointer vers les divers diagrammes et textes figurant dans les résultats de la phase d'analyse. Le pointeur peut être défini en termes de référence aux sections appropriées.

Un processus itératif peut être nécessaire pour assurer la traçabilité jusqu'aux besoins au niveau objet considéré dans la première phase, à partir de la phase de conception. Ce processus est nécessaire car les résultats des phases sont définis relativement à différents niveaux de détail.

NOTE – Tous les besoins ne seront pas traçables dans la réalisation. Ainsi, des besoins telles la disponibilité, la redondance, etc., peuvent ne pas être reflétés dans une réalisation d'interface particulière même s'ils doivent être pris en considération dans une implémentation. Il n'existe pas de mécanisme formel défini dans la présente Recommandation UIT-T garantissant la traçabilité des besoins entre les trois phases. Une approche consiste à ajouter des renvois aux paragraphes et sous-paragraphes dans les résultats des phases besoins et analyse pendant la phase de conception.

## **2.6 Structure de la documentation**

Même s'il y a trois phases, la documentation de l'interface peut associer les résultats de ces phases en un ou plusieurs documents. Il est recommandé de combiner les besoins et l'analyse et d'élaborer des documents de conception distincts pour chaque paradigme spécifique protocolaire de gestion de réseau.

## **Directives pour la définition de l'interface de gestion (GDMI)**

### **A.1 Introduction**

On trouvera ci-après les directives pour la définition de l'interface de gestion (GDMI). Le gabarit GDMI définit la structure pour spécifier les résultats des phases de besoins, d'analyse et de conception.

### **A.2 Gabarit GDMI**

#### **A.2.1 Domaine d'application**

Définit les objectifs principaux et les interfaces du RGT applicables (et les points de référence) de cette spécification. Il utilise le classement catégorie de la Recommandation UIT-T M.3200 [2] pour identifier le ou les service(s) de gestion pris en charge par cette interface.

Ce sous-paragraphe doit donner une description claire des avantages procurés aux utilisateurs du RGT, c'est-à-dire les raisons motivant ce service de gestion. Le contexte doit également être précisé si nécessaire, mais la partie explicative et descriptive doit être distincte. L'information contextuelle sera alors insérée dans un appendice.

#### **A.2.2 Spécifications**

##### **A.2.2.1 Spécifications professionnelles**

Donner la liste des principales spécifications sous forme de texte et identifier les cas d'utilisation avec les acteurs/rôles et les ressources. Le cas d'utilisation doit faire apparaître les besoins de haut niveau et est distinct de la spécification des besoins en ce sens qu'il ne précise pas les niveaux inférieurs. Les informations relatives à la politique (par exemple sécurité, persistance) peuvent être incluses à ce niveau. Pour des raisons de traçabilité, il est recommandé de numéroter les besoins.

##### **A.2.2.1.1 Rôles d'acteur**

Une description textuelle de l'acteur est incluse ici.

##### **A.2.2.1.2 Ressources de télécommunication**

Une description textuelle des ressources de télécommunication requises pour prendre en charge les cas d'utilisation est incluse ici.

##### **A.2.2.1.3 Cas d'utilisation de haut niveau**

Le diagramme de cas d'utilisation de haut niveau est présenté. Afin que les experts puissent comprendre les cas d'utilisation, ces diagrammes doivent être complétés par une description textuelle pour chaque cas d'utilisation. La description doit servir à deux fins: intégrer les connaissances des experts du domaine et valider les modèles dans les phases d'analyse et de conception relativement aux besoins. Un exemple d'un cas d'utilisation de haut niveau est donné dans l'Appendice I.

Les cas d'utilisation de haut niveau peuvent identifier différents ensembles de fonctions définis dans la Recommandation UIT-T M.3400 [3]. Ces cas d'utilisation peuvent être encore plus détaillés comme décrit dans le sous-paragraphe spécification des besoins ci-après en utilisant les stéréotypes tels que "inclusion" et "extension".

Si nécessaire, on peut utiliser des diagrammes séquentiels et d'état. Toutefois, il n'est pas prévu d'utiliser ces diagrammes pour les besoins de haut niveau. Lorsque les codes d'utilisation à ce niveau sont ensuite décomposés dans le niveau suivant des besoins, ces diagrammes peuvent s'avérer plus utiles.

La traçabilité du niveau suivant des besoins à partir de ce niveau est peut-être identifiée par la façon dont chaque ensemble de fonctions est détaillé avec de nouveaux cas d'utilisation.

### **A.2.2.2 Spécification des besoins**

Les cas d'utilisation à haut niveau sont de nouveau détaillés au moyen des fonctions de gestion extraites de la Recommandation UIT-T M.3400. Étant donné que la Recommandation UIT-T M.3400 n'est pas suffisamment exhaustive pour couvrir tous les services de gestion pour tous les domaines de gestion, de nouvelles fonctions seront probablement nécessaires. Ces nouvelles fonctions doivent être incluses comme indiqué ci-dessous.

#### **A.2.2.2.1 Rôles d'acteur**

Une liste de tous les acteurs et une description textuelle de tous les acteurs non encore définis dans les spécifications de haut niveau sont incluses ici.

#### **A.2.2.2.2 Ressources de télécommunication**

Une liste de toutes les ressources passives et une description textuelle de toutes les ressources non encore définies dans les spécifications de haut niveau sont présentées ici.

#### **A.2.2.2.3 Fonctions de gestion RGT**

Les fonctions de gestion identifient les interactions entre les différents rôles d'acteur. Les besoins peuvent inclure un ou plusieurs des éléments suivants: cas d'utilisation, diagrammes séquentiels, diagrammes d'état pour diverses fonctions dans le domaine problème et des descriptions textuelles.

#### **A.2.2.2.4 Cas d'utilisation**

Un exemple d'ajustement de diagrammes de cas d'utilisation de haut niveau est présenté dans l'Appendice I. L'ajustement est obtenu en utilisant les stéréotypes "extension" et "inclusion".

Si nécessaire, on pourra utiliser des diagrammes séquentiels et des diagrammes d'état.

### **A.2.2.3 Analyse**

L'analyse inclut la décomposition fonctionnelle, les flux informationnels, les diagrammes de classe (y compris les relations entre classes), les diagrammes séquentiels et les diagrammes/tableaux d'état. Les diagrammes de classe peuvent être complétés par des détails sur les attributs et opérations autorisées. Comme dans tout diagramme UML, une description textuelle est requise pour compléter les figures. L'inclusion de tous les attributs ainsi que leurs propriétés au sein d'un diagramme de classe peut altérer la lisibilité. Pour une question de présentation, il est permis de ne pas représenter des attributs dans tous les diagrammes de classe.

Les descriptions détaillées des fonctions de gestion et des interactions entre les fonctions doivent être fournies.

Le flux informationnel associé à chaque fonction doit généralement être indiqué en utilisant des tableaux simples définissant les flux. Les tableaux doivent préciser si l'information a un caractère obligatoire, optionnel ou conditionnel. La condition doit être définie. L'analyse peut inclure des modèles d'état comme résultat du flux informationnel. Les transitions d'état peuvent être décrites au moyen de tableaux ou de diagrammes identifiant les événements et l'état résultant. Un autre domaine à prendre en considération lorsqu'on documente l'analyse des besoins concerne les cas d'exception/d'erreur.

Les scénarios décrivant le flux d'information parmi les entités peuvent être décrits en utilisant des diagrammes séquentiels comme cela est illustré dans l'Appendice I.

Le diagramme séquentiel est un diagramme de flux de messages dans lequel l'ordonnée représente le temps et l'abscisse la circulation des messages. Ce diagramme modélise le gestionnaire qui interagit

avec le système géré via le modèle informationnel de gestion. Dans le diagramme se trouvent des instances d'objet qui collaborent par transfert de messages pour exécuter une tâche fonctionnelle donnée.

Les conditions préalables et les postconditions peuvent être utilisées pour décrire les flux d'information dans les diagrammes d'interaction.

Les numéros de référence doivent être assignés aux sections analyse et peuvent être utilisés pour permettre la traçabilité depuis le modèle ou le référençage depuis le modèle selon le cas.

La traçabilité doit être assurée pour les différents diagrammes et textes avec la phase besoins au moyen de renvois aux cas d'utilisation.

#### **A.2.2.4 Conception**

Les modèles informationnels propres au paradigme de protocole sont présentés dans la présente section (par exemple, GDMO/ASN.1, IDL).

## **ANNEXE B**

### **Règles d'affectation des identificateurs d'objet RGT**

On applique la procédure d'affectation d'identificateurs d'objet suivante en cas d'utilisation du protocole CMIP et des directives GDMO pour la conception d'une interface RGT.

#### **B.1 Structure des identificateurs d'objet RGT**

L'Annexe C/X.680<sup>1</sup> [6] définit les tout premiers arcs de la structure des identificateurs d'objet à utiliser pour les éléments d'information dans les Recommandations de l'UIT-T. Tous les identificateurs d'objet ont la structure indiquée sur la Figure B.1 qui constitue une représentation graphique de l'information suivante:

(0) itu-t<sup>2</sup>  
    (0) recommandation  
    (1) a  
    (2) b  
    (3) c  
    .....  
    (7) g  
    (774) g774  
    .....  
    (13) m  
    (3100) m3100  
    .....  
    (14) n  
    .....

Par exemple, l'identificateur d'objet de la Recommandation UIT-T M.3100 est le suivant:

**{ itu-t(0) recommandation(0) m(13) m3100(3100) }**

---

<sup>1</sup> L'Annexe C/X.208 fournit les définitions équivalentes.

<sup>2</sup> Dans la Recommandation X.208 (ASN.1), on a utilisé le sigle "ccitt" pour élaborer la hiérarchie des identificateurs d'objet. Dans les nouvelles Recommandations, il convient d'utiliser "uit-t" qui est synonyme de "ccitt".

Les feuilles de la structure ci-dessus représentent des Recommandations de l'UIT-T. La sous-structure RGT suivante doit être utilisée en dessous de chaque feuille représentant une Recommandation. Cette sous-structure est établie d'après les règles définies dans la Recommandation UIT-T X.722.

- (0) **informationModel**
  - (0) **standardSpecificExtension**
  - (2) **asn1Module**
  - (3) **managedObjectClass**
  - (4) **package**
  - (5) **parameter**
  - (6) **nameBinding**
  - (7) **attribute**
  - (8) **attributeGroup**
  - (9) **action**
  - (10) **notification**
  - (11) -- *les deux nœuds suivants sont réservés pour utilisation future avec le GRM*
  - (12)
- (1) **protocolSupport**
  - (0) **applicationContext**
- (2) **managementApplicationsSupport**
  - (0) **standardSpecificExtension**
  - (1) **functionalUnitPackage**
  - (2) **asn1Module**
- (127) **dot** -- *pour les parties d'une Recommandation (voir B.2)*

Il est recommandé que les références aux valeurs soient définies comme suit dans le cadre d'un module ASN.1 pour les feuilles de la sous-structure RGT précitée, par exemple pour **managedObjectClass**:

```
<recommendation>ObjectClass OBJECT IDENTIFIER
    ::= { itu-t(0) recommendation(0) <recommendation series letter>(number)
        <recommendation>(number) informationModel(0)
        managedObjectClass(3) }
```

Exemple:

```
m3100ObjectClass OBJECT IDENTIFIER
    ::= { itu-t(0) recommendation(0) m(13) m3100(3100) informationModel(0)
        managedObjectClass(3) }
```

Pour l'information de gestion à communiquer ou réutilisable dans d'autres gabarits, il faut enregistrer le gabarit qui définit cette information. Chaque gabarit d'information de gestion à enregistrer est identifié par un identificateur d'objet.

A titre d'exemple, une classe d'objets appelée **exampleObjectClass** dans la Recommandation UIT-T M.3100 aura l'identificateur d'objet suivant:

```
exampleObjectClass MANAGED OBJECT CLASS
    .
    .
    .
    REGISTERED AS { m3100ObjectClass 5 };
```

Il convient d'appliquer la même méthode pour les autres feuilles de la sous-structure RGT.

Dans le paragraphe "abréviations" de la Recommandation, inclure les références aux valeurs ainsi que les séquences de valeurs de l'identificateur d'objet de cette référence à une valeur, par exemple:

```
m3100ObjectClass { itu-t(0) recommendation(0) m(13) m3100(3100)
    informationModel(0) objectClass(3) }
```

## B.2 Extension de la structure des identificateurs d'objet RGT aux "parties" de Recommandation

La structure décrite au B.1 doit également être appliquée aux Recommandations qui utilisent des parties numérotées (voir ci-après):

(0) itu-t

(0) recommendation

(1) a

(2) b

(3) c

.....

(7) g

(774) g774

(127) dot

(1) part1

.....

(13) m

(3100) m3100

.....

(14) n

.....

Par exemple, l'identificateur d'objet pour la Recommandation UIT-T G.774.1 est:

```
{ itu-t(0) recommendation(0) g(7) g774(774) dot(127) part1(1) }
```

La sous-structure en dessous de ce niveau est celle qui est définie dans la Recommandation UIT-T X.722 et décrite au B.1 ci-dessus.

Exemple:

```
g774dot1ObjectClass OBJECT IDENTIFIER ::=
```

```
{ itu-t(0) recommendation(0) g(7) g774(774) dot(127) part1(1) informationModel(0)
  managedObjectClass(3) }
```

Ainsi, une classe d'objets appelée `exampleObjectClass` dans la Recommandation UIT-T G.774.1 aura l'identificateur d'objet suivant:

```
exampleObjectClass MANAGED OBJECT CLASS
```

.

.

.

```
REGISTERED AS { g774dot1ObjectClass 5 };
```

Les références tirées d'autres Recommandations doivent avoir le format suivant:

```
"Recommandation G.774.1: 1994"
```

## B.3 Procédures d'affectation appliquées au RGT

Il est recommandé d'appliquer les procédures d'affectation suivantes:

- 1) un élément d'information de gestion se voit affecter un seul identificateur d'objet et est défini dans un seul document. Si un élément d'information de gestion est requis dans une Recommandation et qu'il est déjà défini ailleurs, on utilise une référence au gabarit existant. La référence à un gabarit identifie la Recommandation et la date publiée avec l'étiquette-gabarit, par exemple "Recommandation M.3100: 1992": exemple de classe d'objets. Chaque Recommandation doit également comporter la déclaration suivante:

"Lorsque d'autres documents font référence aux définitions des gabarits de la présente Recommandation, il convient d'utiliser le préfixe, par exemple "Recommandation M.3100: 1992", pour identifier la source des définitions.";

- 2) chaque Commission d'études est responsable (autorité d'enregistrement) de l'enregistrement des identificateurs d'objet des Recommandations qu'elle élabore et qui sont situées dans les arcs de la sous-structure du RGT définie plus haut.

#### B.4 Affectation des identificateurs d'objet dans un contexte d'application RGT

L'identificateur d'objet ci-après dans un contexte d'application RGT est défini et enregistré dans la Recommandation UIT-T M.3100 et doit être utilisé par toutes les applications RGT:

La valeur d'identificateur d'objet:

```
{ itu-t(0) recommendation(0) m(13) m3100(3100) protocolSupport(1)
  applicationContext(0) tmnApplicationContextOne(1) }
```

est assignée au contexte d'application qui possède les mêmes capacités que le contexte d'application des systèmes dans la Recommandation UIT-T X.701 mais prend en compte également les valeurs entières de ProbableCause (cause probable) comme cela est défini dans la Recommandation UIT-T M.3100.

La Figure B.1 décrit la partie "supérieure" de la structure d'un identificateur d'objet RGT.

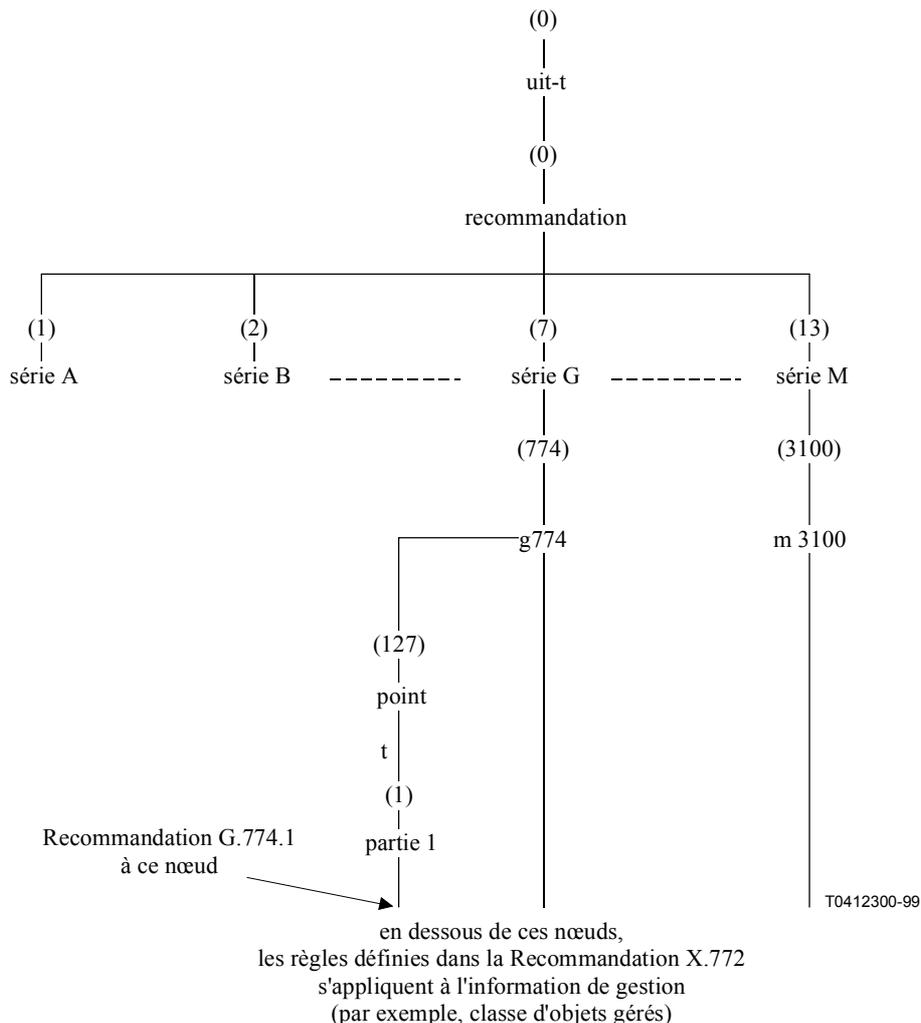


Figure B.1/M.3020 – Représentation graphique de l'arbre d'identificateurs d'objet

## Opérations génériques

### C.1 Opérations génériques

#### C.1.1 getAttributes

L'opération générique `getAttributes` (dans `attributeNameList: AttributeNameListType`, `out attributeNameValuePairList: AttributeNameValuePairListType`) dans laquelle la liste de noms (`NameList`) d'attribut de paramètres est utilisée pour indiquer toute combinaison des attributs (soumis à d'éventuelle contraintes supplémentaires) qui doivent être retournés dans la liste `attributeNameValuePairList` du paramètre.

#### C.1.2 getAllAttributes

Cette opération générique (`out attributeNameValuePairList: AttributeNameValuePairListType`) est utilisée pour extraire des valeurs de tous les attributs d'une classe. La réponse est renvoyée dans la liste `attributeNameValuePairList`.

#### C.1.3 notifications

Pour spécifier les notifications, le stéréotype `<<Distribution de notification>>` a été défini. Il est représenté sous forme de diagramme à l'aide du symbole utilisé pour les classes UML. La signature d'une ou plusieurs notifications est spécifiée sous forme d'opérateur dans un stéréotype `<<Distribution de notification>>` (`<<NotifyDispatch>>`). La sémantique est la suivante: les opérations dans un compartiment `<<Distribution de notification>>` sont utilisées par les classes UML pour déclencher la distribution des notifications via un certain mécanisme de distribution d'événements qui est plus complètement spécifié dans la phase conception.

Le déclenchement de plusieurs opérations de notification dans un compartiment `<<Distribution de notification>>` implique la capacité à envoyer tous les types de notification indiqués.

La Figure C.1 montre un exemple de spécification de deux stéréotypes de distribution de notification. L'exemple d'alarme a un type de notification comme `alarm`, qui est une signature de paramètre qui est une simplification de la syntaxe `commAlarm` de la Recommandation UIT-T X.733. L'exemple de `configEvents` a quatre types de notifications qui ont des signatures de paramètre qui sont des simplifications de la syntaxe définie dans les Recommandations UIT-T X.730 et X.731.

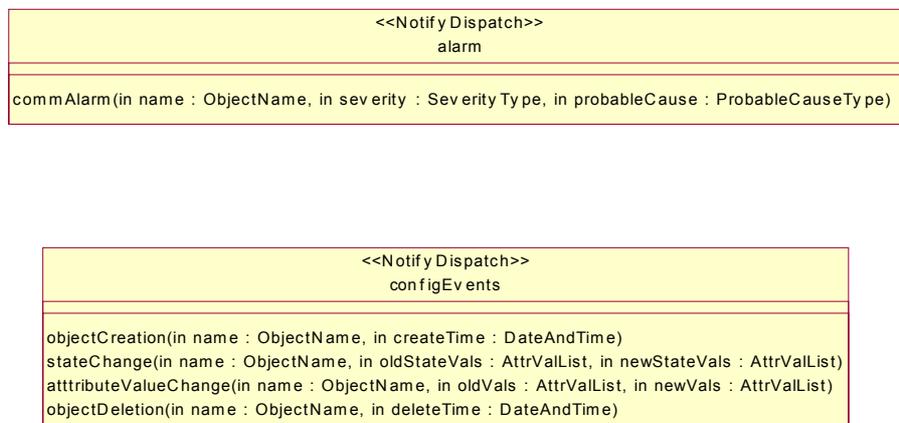
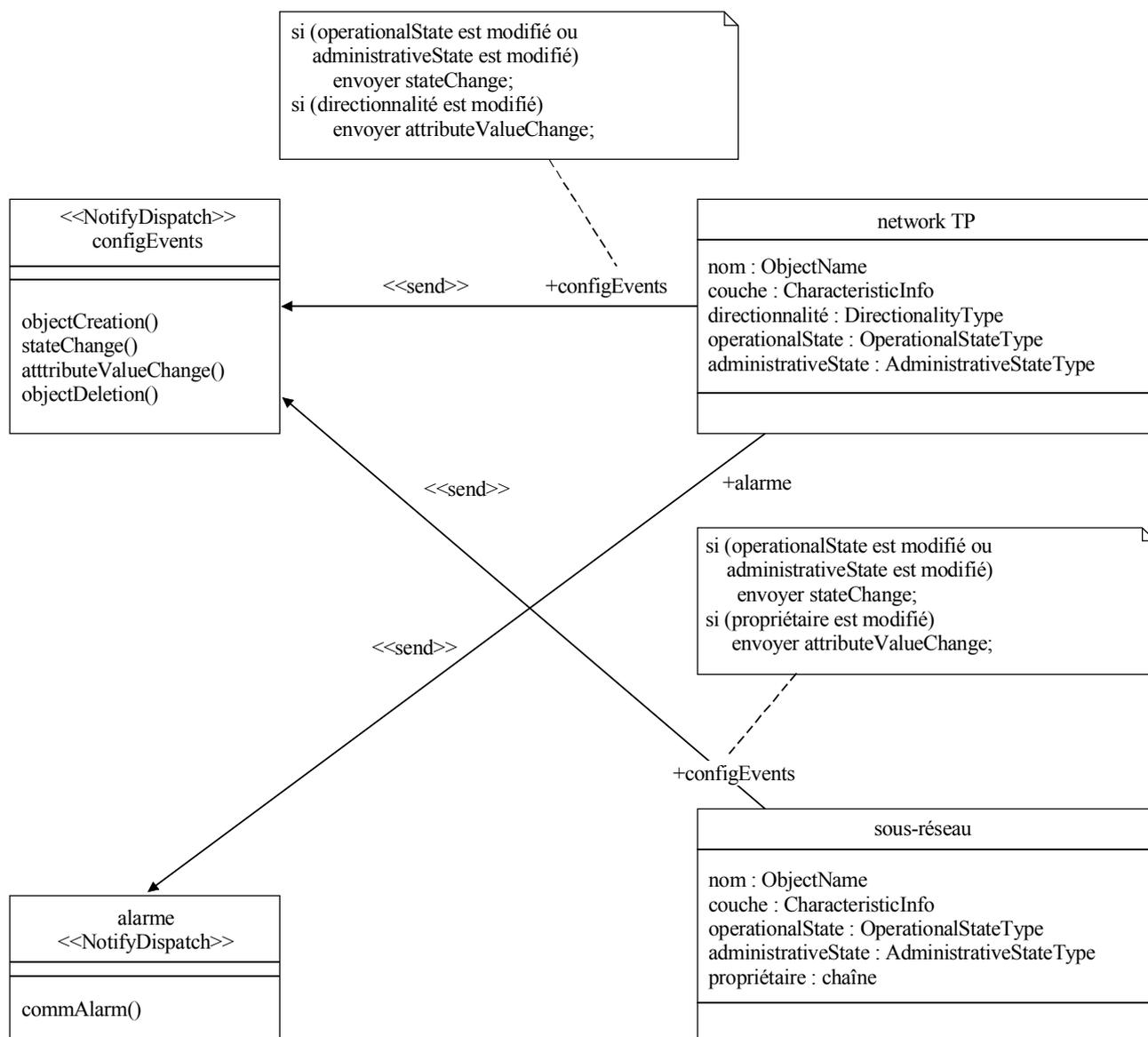


Figure C.1/M.3020 – Spécification des ensembles de notifications génériques

La Figure C.2 montre un exemple de la façon d'utiliser le stéréotype <<send>> de l'élément standard association UML pour spécifier qu'une instance d'une classe doit être capable d'envoyer des notifications. Lorsqu'une classe a une association <<send>> avec un stéréotype <<NotifyDispatch>>, cela implique que la classe doit être capable d'envoyer tous les types de notification indiqués dans le compartiment opération du stéréotype <<NotifyDispatch>> associé. Pour la lisibilité, des abréviations pour les signatures d'opération de notification (c'est-à-dire une liste des paramètres vides) peuvent être utilisées, comme le montre la Figure C.2.

Le rôle de la fin d'association sur le côté de la classe peut avoir des contrats qui spécifient les conditions sous lesquelles les types de notification spécifiques doivent être émis par les instances de cette classe.



T0412310-99

**Figure C.2/M.3020 – Spécification des notifications émises par classes**

## APPENDICE I

### Exemple d'utilisation des directives GDMI (fourniture de LCS)

#### I.1 Introduction

Le présent appendice contient un exemple d'utilisation d'un gabarit GDMI. Cet exemple montre comment les phases de spécification, d'analyse et de conception peuvent être documentées en associant du texte, des diagrammes UML et une représentation tabulaire. Cet exemple est fondé sur la Recommandation UIT-T M.3208.1 [5] pour la fourniture d'un service de circuits loués spécialisés. L'utilisation du langage UML dans les trois phases et la façon dont la phase de conception peut renvoyer à des définitions de comportement dans la phase analyse sont illustrées ici. Cet exemple ne couvre qu'un petit ensemble des spécifications contenues dans la Recommandation UIT-T M.3208.1. Au besoin, on utilise le texte actuel de la Recommandation UIT-T M.3208.1.

#### I.2 Gabarit GDMI

##### I.2.1 Domaine d'application

La présente Recommandation UIT-T décrit un sous-ensemble de services de gestion du RGT pour le réseau de circuits loués spécialisés identifié dans la Recommandation UIT-T M.3200 comme étant un domaine géré RGT. Elle met principalement l'accent sur les services de gestion d'administration des clients et de gestion de la maintenance pour les services de circuits loués point à point (LCS, *leased circuit service*) qui peuvent être proposés par un ou plusieurs fournisseurs de services et être placés sous le contrôle du SC (client du service) avec différents niveaux de visibilité. Les services LCS sont définis entre un seul SC et un seul SP (fournisseur du service). Ces services de gestion sont également applicables aux interactions entre les systèmes de gestion de différents fournisseurs de services ou à l'intérieur d'un même fournisseur de services.

##### I.2.2 Besoins

###### I.2.2.1 Besoins professionnels

Les services de gestion du RGT dans la présente Recommandation UIT-T spécifient les interfaces pour des services de circuits loués entre un système d'exploitation (OS, *operations system*) et un système d'exploitation (OS) pour fournir et gérer des services de circuits loués. Les interfaces visées par les services de gestion du RGT dans la présente Recommandation UIT-T sont applicables à la fois aux interfaces X au-delà des frontières juridiques et des interfaces Q à l'intérieur d'un RGT. La prise en charge de ces services décrits dans la présente Recommandation UIT-T est laissée à la discrétion du fournisseur de services.

En général, la définition d'un service doit être indépendante du réseau utilisé pour acheminer le service. Ceci permet d'utiliser des technologies différentes pour prendre en charge le service. Par conséquent, l'information au niveau du réseau ne doit pas être présentée à la couche service. Toutefois, les caractéristiques spécifiques du service peuvent être définies, ce qui permet de présenter à un client du service l'information relative au réseau ou à l'élément de réseau. Dans ce cas, une abstraction de l'information convenant à cette caractéristique de service est transférée.

###### I.2.2.1.1 Rôles d'acteur

###### Clients du service

Client du service; voir la définition du terme "client" dans la Recommandation UIT-T M.3320. Cette utilisation de l'expression client du service spécialise la définition sur le contexte du rôle de gestion RGT pour le niveau service.

## Fournisseurs de services

Référence générale à une entité qui fournit des services de télécommunication à des clients et à d'autres utilisateurs soit sur la base d'un tarif ou sur contrat. Il n'est pas nécessaire qu'un fournisseur de services exploite un réseau. Un fournisseur de services peut ne pas être un client d'un autre fournisseur de services. Dans le présent appendice, l'expression "(sous-)réseau du SP" est utilisée pour indiquer le ou les réseaux utilisés par le SP pour fournir le service LCS.

### I.2.2.1.2 Ressources de télécommunication

#### Service de circuits loués spécialisés

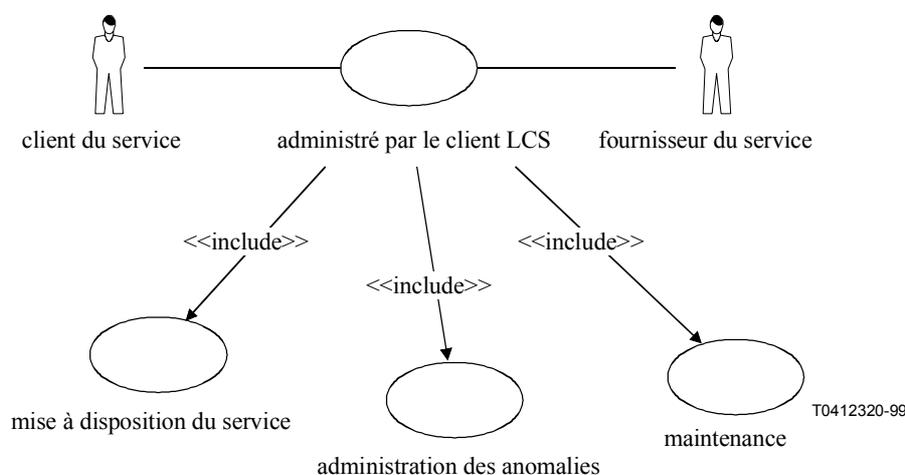
Ce service est une connexion point à point entre deux points d'accès au service qui ne peuvent être modifiés après la création du service. Le service de circuits loués spécialisés utilise le nom de service et la classe de service pour définir la valeur des paramètres propres au service et désigner quels paramètres peuvent être modifiés après la mise à disposition du service.

### I.2.2.1.3 Cas d'utilisation de haut niveau

Au niveau le plus élevé, la zone gérée pour la Recommandation UIT-T M.3200 est "l'administration par le client du service de circuits loués". Un acteur dans le rôle du client du service interagit avec un acteur dans le rôle de fournisseur de services pour exécuter les diverses activités de CNM. Ces activités sont regroupées au moyen d'ensemble de fonctions définies dans la Recommandation UIT-T M.3400 où dans de nouvelles définitions pour disposer de capacités additionnelles. Un fournisseur de services peut assumer le rôle de client du service si le service de bout en bout doit être fourni et maintenu par plusieurs fournisseurs de services.

La Figure I.1 est le cas d'utilisation de plus haut niveau dans lequel un acteur client du service interagit avec un acteur fournisseur du service. Le cas d'utilisation LCS administré par le client fait appel aux ensembles de fonctions indiquées par ces trois cas d'utilisation. La présente Recommandation UIT-T traite des besoins et de l'analyse correspondante aux aspects mise à disposition du service. D'autres Recommandations de la série étendent les sujets abordés ici aux groupes d'ensembles de fonctions.

Les cas d'utilisation représentant des groupes d'ensembles de fonctions sont précisés et détaillés en ensembles de fonctions et finalement en fonctions de gestion. Comme on le verra plus loin, les fonctions dans un cas d'utilisation d'ensemble de fonctions peuvent étendre les fonctions dans un autre ensemble pour respecter tous les besoins du service de gestion.

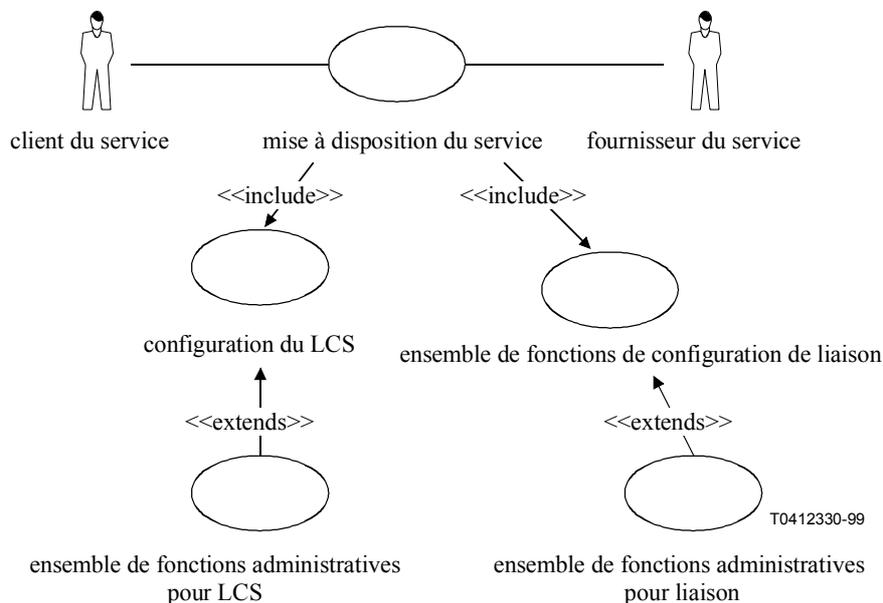


**Figure I.1/M.3020 – Cas d'utilisation administration par le client d'un service LCS**

L'acteur client du service interagit avec l'acteur fournisseur du service pour répondre aux besoins correspondant au cas d'utilisation appelé "mise à disposition du service". Le cas d'utilisation "mise à disposition du service" utilise deux cas d'utilisation: l'ensemble de fonctions de configurations LCS et l'ensemble de fonctions de configurations de liaison. La mise à disposition du service de cas d'utilisation décrit les besoins pour l'acteur client du service afin de demander la création d'un service de circuits loués ou de connexion de liaison préapprovisionnée. Ce dernier cas peut être utilisé par le client du service (par exemple) pour créer des services de circuits loués en temps réel en choisissant des connexions par liaisons spécifiques à connecter (éventuellement pour un grand événement sportif programmé) pendant un certain temps et pendant certaines périodes de certains jours. Le stéréotype <<include>> est utilisé pour indiquer que la mise à disposition du service pour le cas d'utilisation utilise les fragments réutilisables pour créer des circuits loués et des connexions de liaison.

Les activités de création d'un cas d'utilisation d'ensemble de fonctions de configuration LCS sont élargies par le cas d'utilisation ensemble de fonctions administratives. Le stéréotype <<extends>> est utilisé pour indiquer ceci comme suit: l'ensemble de fonctions de configuration contient des activités pour le client du service afin de fournir les capacités minimales pour l'ordre de création de service (demande de service). Lorsque la création du service n'est pas possible en temps réel, les fonctions administratives sont utilisées par le fournisseur du service. Le fournisseur du service peut donner des informations par exemple sur la progression de la demande (par exemple une date de disponibilité pour le service a été étendue ou la largeur de bande demandée ne peut pas être fournie, etc.). Le stéréotype <<extends>> fait que le cas d'utilisation procède aux fonctions dans l'ensemble de fonctions de configuration LCS en fournissant (par exemple) des capacités de signalement sur l'état de la demande pour créer un nouveau service ou pour modifier une demande de service précédemment formulée ou un service existant.

La Figure I.2 décrit les cas d'utilisation auxquels il est demandé de satisfaire les besoins du client du service pour les activités de mise à disposition du service.



**Figure I.2/M.3020 – Cas d'utilisation de mise à disposition du service dans le cas de la CNM de service LCS et de connexions de liaison**

### **I.2.2.2 Spécification proprement dite**

L'ensemble de fonctions de configuration LCS, tel que noté ci-dessus, décrit les scénarios permettant à un acteur client du service d'interagir avec un acteur fournisseur du service pour demander la mise à disposition d'un service LCS (en temps réel ou non) ou une connexion de liaison (en temps non réel). La spécialisation de l'ensemble de fonctions génériques de mise à disposition dans la Recommandation UIT-T M.3400 pour les services LCS et les connexions de liaison permet d'affiner encore plus le cas d'utilisation.

#### **I.2.2.2.1 Rôles d'acteur**

Aucun nouveau rôle d'acteur hormis ceux identifiés dans la partie spécification professionnelle n'est nécessaire pour la spécification proprement dite.

#### **I.2.2.2.2 Ressources de télécommunication**

Aucune autre ressource hormis celle identifiée dans la partie spécification professionnelle n'est nécessaire.

#### **I.2.2.2.3 Fonctions de gestion RGT**

Le code d'utilisation d'ensemble de fonctions de configuration LCS comporte trois fonctions: demande de création d'un service, suppression d'un service existant ou modification des paramètres d'une demande création de service précédemment formulée ou d'un service existant. Par définition, une demande de création d'un service est une commande de services, le client du service passant une commande de service à l'intention du fournisseur de services. Une fois que la commande de service est créée, un acteur client du service peut annuler la commande passée.

NOTE – Même si l'exemple montre que l'annulation de la demande est associée à une fonction de création de service seulement, elle peut être utilisée pour élargir les deux autres fonctions – annulation de la demande de suppression ou de la demande de modification. L'accord sur le niveau du service (appelé SLA ou contrat) entre le fournisseur du service et le client du service définit les politiques associées à cette annulation. Parmi les exemples de décisions politiques: si les activités ont déjà commencé sur la base de la demande quelle est la politique de comptabilité pour la partie déjà exécutée, et les exigences de sécurité associées avec l'annulation d'une demande.

#### **I.2.2.2.4 Cas d'utilisation**

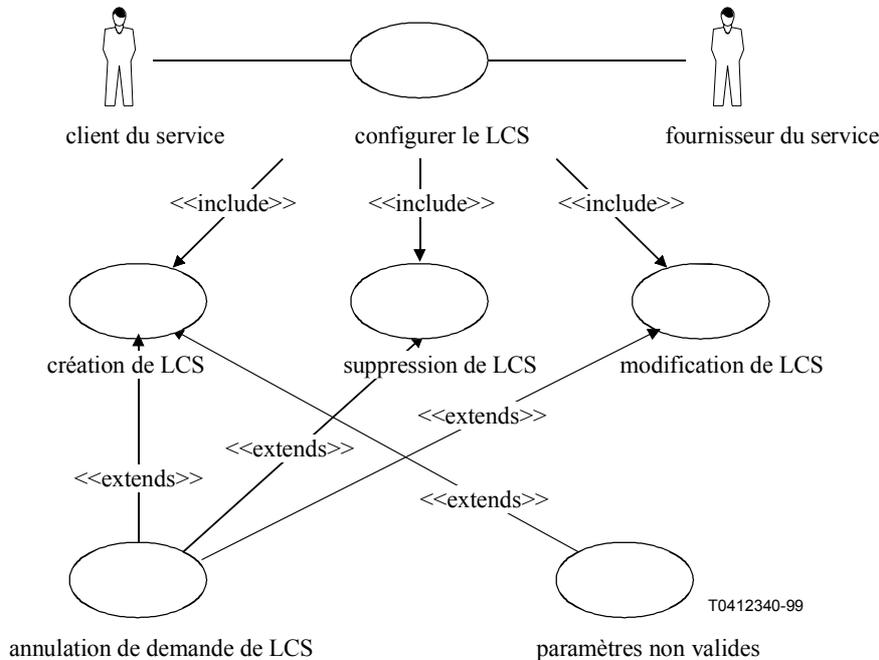
La Figure I.3 illustre l'ensemble de fonctions de configuration LCS en termes de cas d'utilisation suivants: création de LCS, suppression de LCS et modification de LCS. Le fournisseur du service interagit avec le cas d'utilisation d'ensemble de fonctions de configuration qui utilise les trois cas d'utilisation. Le cas d'utilisation création de LCS définit les exigences de la part d'un client du service pour émettre une commande de service demandant le nom, la classe ainsi que les valeurs des paramètres intéressés pour le service demandé.

Le cas d'utilisation demande d'annulation de LCS est élargi en définissant le scénario dans lequel un client peut annuler une demande de service précédemment formulée. Si la demande contient des paramètres non valides (par exemple la classe du service demandé ou le nom de ce service n'est pas valide ou n'est pas proposé(e) par le fournisseur du service), il apparaît une condition d'erreur et des scénarios d'erreur appropriés sont générés par le cas d'utilisation étendue "paramètres de demande non valides".

NOTE – De nouveaux stéréotypes peuvent être nécessaires pour la définition des besoins dans certains services de gestion et domaines de gestion. Cet exemple utilise seulement des stéréotypes définis en UML.

Pour la spécification proprement dite, les fonctions CNM sont identifiées. Le sommaire associé à ces fonctions peut être inclus ici. A titre d'exemple, la description modifiée suivante reprise de la Recommandation UIT-T M.3208.1 peut être utilisée pour le cas d'utilisation création de LCS:

Ce cas d'utilisation permet au client du service de demander la création d'un ou de plusieurs services de circuits loués spécialisés. Le client du service doit identifier le service à fournir, les caractéristiques du service (telles que spécifiées dans le flux d'information), la date demandée de disponibilité du service, le contact du client au sein de l'organisation et les informations utiles concernant les points de départ et de terminaison du service (voir le flux d'information). Le client du service peut également spécifier le trajet du service demandé et un identificateur d'utilisateur pour le circuit loué demandé. Le fournisseur du service peut rejeter la demande (décrite dans le cas d'utilisation étendue) si le fournisseur du service n'offre pas le service demandé.



**Figure I.3/M.3020 – Décomposition du cas d'utilisation ensemble de fonctions de configuration LCS**

Les cas d'utilisation doivent satisfaire à tous les besoins couverts dans les fonctions décrites au I.2.2.2.3.

### I.2.2.3 Analyse

#### I.2.2.3.1 Classes d'objets et diagrammes d'état

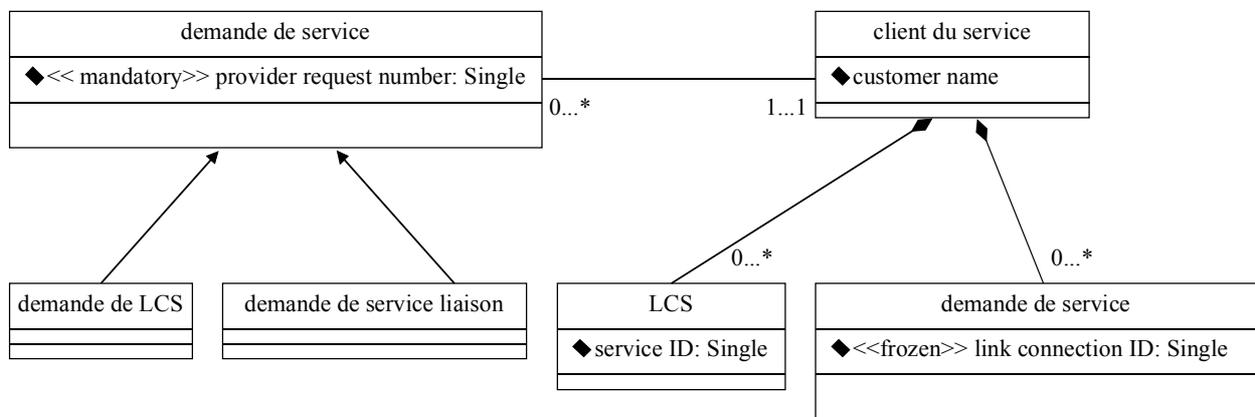
Les classes permettant la prise en charge des cas d'utilisation création de LCS et connexion de liaison sont représentées à la Figure I.4. Une classe d'objets client du service peut formuler une ou plusieurs demandes de service comme indiqué par l'association dont la cardinalité est marquée 1 et \*. En raison de l'existence de propriétés communes associées avec la demande de service LCS et de connexions de liaison, une classe générique appelée "demande de service" est identifiée. Pour améliorer la lisibilité, tous les attributs et une des opérations autorisées sur les classes d'objets ne sont pas représentés ici. L'attribut "numéro de demande du fournisseur" est inclus ici sans montrer la visibilité ou le type (étant donné qu'il s'agit d'attributs qui sont exposés sur une interface pour la gestion, ils ne sont pas considérés comme étant publics, même s'ils peuvent avoir différentes visibilités du point de vue logiciel). Le type n'est pas inclus car il peut être différent en fonction du paradigme de conception. Une relation de généralisation existe entre la demande de service pour créer un service LCS ou une liaison avec la demande de service générique comme cela est montré dans la figure.

La classe de demande de service LCS prend en charge les exigences du cas d'utilisation création de LCS. Les paramètres détaillés sont présentés dans le tableau de flux informationnel. Les interactions entre le client du service et le fournisseur du service pour ce cas d'utilisation sont montrées dans des diagrammes séquentiels de collaboration et d'activité.

Le nom du client est un attribut obligatoire de la classe d'objets client du service. Un client peut disposer de plusieurs services provenant d'un fournisseur de services et on utilise l'agrégation pour décrire cette situation. Le service LCS et la connexion de liaison de l'objet service résultent de l'exécution réussie des cas d'utilisation du service création de LCS et création de connexion de liaison. Un service doit appartenir à un client et la cardinalité entre le client et les services indique cette exigence.

Lorsqu'un service est créé (LCS ou connexion de liaison) il est identifié au moyen de l'attribut "ID de service". La contrainte {gel} est utilisée pour indiquer que, après la création d'un service, la valeur de cet attribut peut ne pas changer durant la persistance de l'objet. Cela implique également que l'attribut ne peut être que lu et qu'il ne peut pas être modifié par le client du service ou par le fournisseur du service. (Cela peut être prévu au moyen de la capacité fixée lors de la création disponible dans le CMIP-GDMO.)

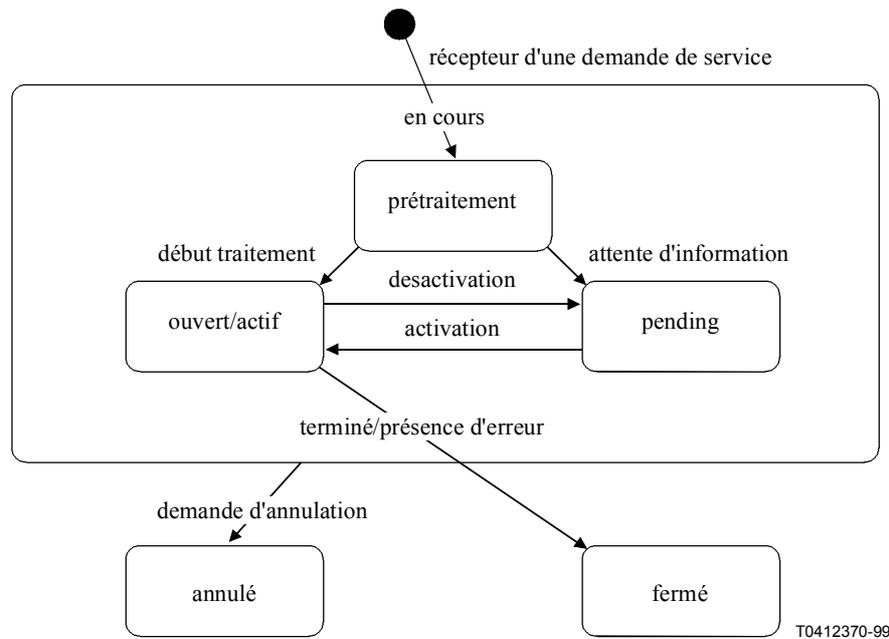
Le diagramme de transition d'état de la Figure I.5 est associé à l'objet demande de service. La définition des états se trouve dans la Recommandation UIT-T M.3208.1 avec certaines différences légères tenant à l'adaptation à la notation UML. Pour éviter la répétition de la description des états, les événements transition ne sont pas décrits dans le présent appendice et le lecteur est invité à se reporter à la Recommandation UIT-T M.3208.1. Afin de montrer que l'annulation peut être formulée lorsque la demande se trouve dans l'un des trois états, un super état "en cours" a été introduit. Sans cet état, le diagramme aurait annulé l'événement depuis chacun des trois états.



T0412360-99

**Figure I.4/M.3020 – Structure de classe pour la création de LCS et connexion de liaison**

La Figure I.5 représente les états et les transitions entre états pour la demande de service.



**Figure I.5/M.3020 – Diagramme d'état pour la classe demande de service**

Le diagramme d'état tient compte des conditions exposées dans le texte qui suit la Figure I.1. Les transitions montrent que la disponibilité du service demandé, si elle n'a pas lieu en temps réel, peut se traduire par le passage à différents états. L'extension assurée par le cas d'utilisation annulation de la demande de LCS est prise en charge par l'événement et l'état appelé "annulé".

### I.2.2.3.2 Diagramme séquentiel

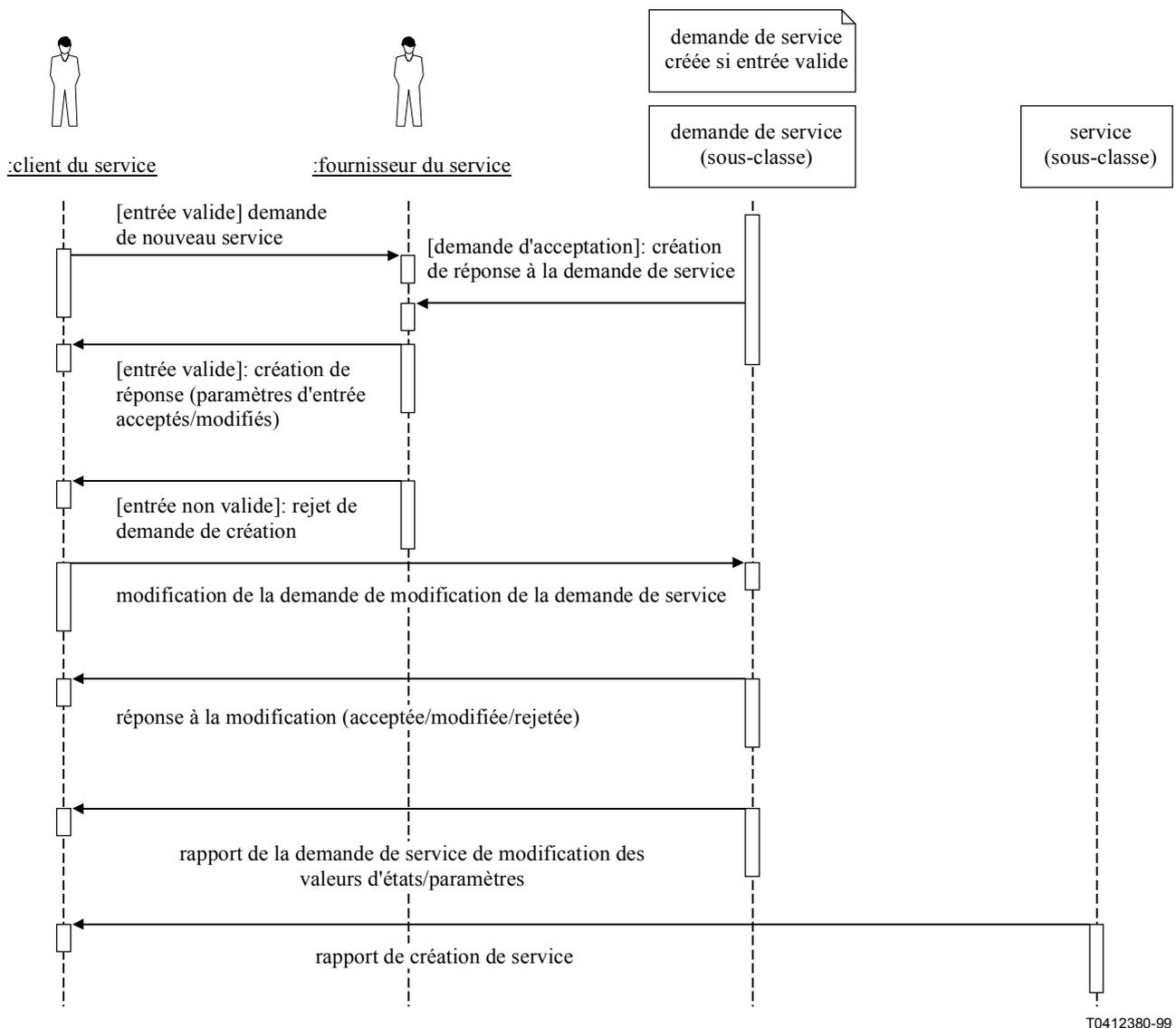
Le diagramme séquentiel de la Figure I.6 représente les séquences de flux de messages lorsqu'un client d'un service envoie une demande de création de service.

NOTE – L'identification de toutes les séquences possibles lors d'une demande de service n'est pas incluse ici. Certains cas illustrant la situation sont représentés. Le diagramme séquentiel inclut les messages qui font partie des différents cas d'utilisation. Certains messages spécifient l'évaluation des conditions et le résultat de cette évaluation détermine si le message est échangé.

Un client du service émet une demande de nouveaux services. Si le fournisseur de services détermine que les paramètres d'entrée sont valides, la demande est acceptée et une sous-classe demande de service est créée. Au lieu de définir un constructeur ou un fabricant, la propriété des notes est utilisée pour expliquer la création de l'objet demande de service si la demande est acceptée par le fournisseur de services. L'acceptation d'une demande de service n'implique pas que toutes les valeurs demandées associées à la création du nouveau service seront disponibles dans le nouveau service. Cela est précisé lorsque les paramètres inclus dans la demande de création sont identifiés ultérieurement dans le détail (voir Tableau I.1). La réponse après la création d'un objet demande inclut les valeurs des paramètres. Comme le montre la Figure I.6, la réponse inclut les valeurs des paramètres qui peuvent être modifiées par rapport à la demande originale. On suppose que le client du service, par défaut, accepte les modifications proposées par le fournisseur du service. Si tel n'est pas le cas, le client peut formuler une annulation de la demande de service. Les politiques associées à l'annulation doivent être disponibles dans un SLA. En outre, sur la base de l'accord de niveau de service, le fournisseur peut conserver un relevé chronologique des modifications demandées. Si le service est créé en temps réel, il n'est pas nécessaire de créer un objet demande de service. Ce cas est traité comme suit: un objet demande de service est créé et supprimé immédiatement et le client du service est avisé du nouveau service créé. Le client du service ne sera pas en mesure d'exécuter la séquence de messages (modification de la demande de service, par exemple) indiquée plus loin.

Après la création de l'objet demande de service, le client du service peut surveiller la progression de la demande, demander des modifications aux paramètres et être informé de la progression. Même si le diagramme séquentiel est utilisé pour représenter le domaine temporel, le rapport de statut est asynchrone et peut être émis par le fournisseur de services de manière asynchrone aussi longtemps que l'objet demande de service est créé.

Après que l'état de l'objet demande de service passe à fermé (ce qui suppose que la création a réussi), une notification est émise par le fournisseur de services indiquant la création du service avec les valeurs pertinentes des paramètres associés. Les messages de signalement correspondent aux fonctions qui sont prises en charge par le cas d'utilisation ensemble de fonctions administratives LCS (qui est ensuite affiné en d'autres cas d'utilisation pour signaler les modifications d'états et de valeurs d'attribut avec la fonction de monitoring).



T0412380-99

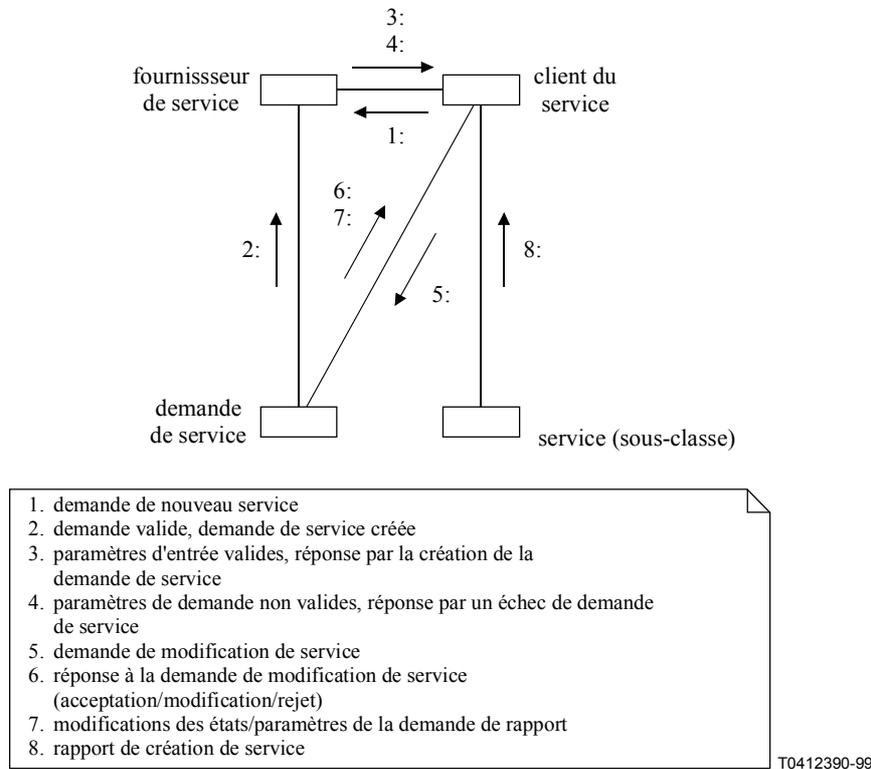
**Figure I.6/M.3020 – Diagramme séquentiel: création effective d'un service**

### I.2.2.3.3 Diagrammes de collaboration

Correspondant au diagramme séquentiel décrivant le flux de messages dans le temps, le diagramme de collaboration pour la création menée à bien d'un service est représenté à la Figure I.7. L'interaction entre les objets et les messages échangés par eux est représentée en ordre séquentiel de

messages dans le diagramme séquentiel. Les événements réels correspondant aux numéros dans la figure sont définis à l'intérieur par des notes.

NOTE – En général, il suffit de montrer le diagramme séquentiel afin d'illustrer la circulation des flux de messages. Dans cet exemple, le diagramme de collaboration est inclus pour montrer que dans certains cas ces diagrammes peuvent inclure des informations supplémentaires et par conséquent, peuvent être nécessaires. Pour ce simple cas, le diagramme de collaboration n'ajoute pas de nouvelles informations par rapport au diagramme séquentiel.



**Figure I.7/M.3020 – Diagramme de collaboration: création effective d'un service**

#### I.2.2.3.4 Flux informationnel

Le diagramme fournit des informations sur les flux de messages à haut niveau. Si les paramètres transmis par les messages doivent être représentés dans le diagramme, ils seront difficiles à lire. Outre la liste des paramètres, il est souvent nécessaire d'identifier si un paramètre est toujours requis dans un échange (obligatoire) ou peut être inclus lorsqu'une condition est remplie à la discrétion de l'utilisateur. Afin d'expliquer les paramètres et les conditions de leur présence ou de leur absence, une approche tabulaire est utilisée. Le Tableau I.1 est extrait sous forme simplifiée de la Recommandation UIT-T M.3208.1. En fonction de la complexité de l'application, tel tableau peut montrer les détails qu'il n'est pas possible de montrer dans un diagramme. La convention pour le "m", etc., est définie dans la Recommandation UIT-T M.3208.1.

Un deuxième avantage de la création d'un tel tableau est la réutilisation de la définition dans un autre service de gestion pour la zone gérée. Par exemple, le service appelé "gestion de connexion" peut être utilisé par un client pour créer un service LCS en temps réel. Dans ce cas, la plupart des paramètres pour la création d'un circuit loué sont communs avec le cas en temps non réel. Certaines restrictions et l'extension des paramètres peuvent être nécessaires. L'utilisation du même tableau et l'explication des contraintes facilitent la réutilisation des définitions de paramètre.

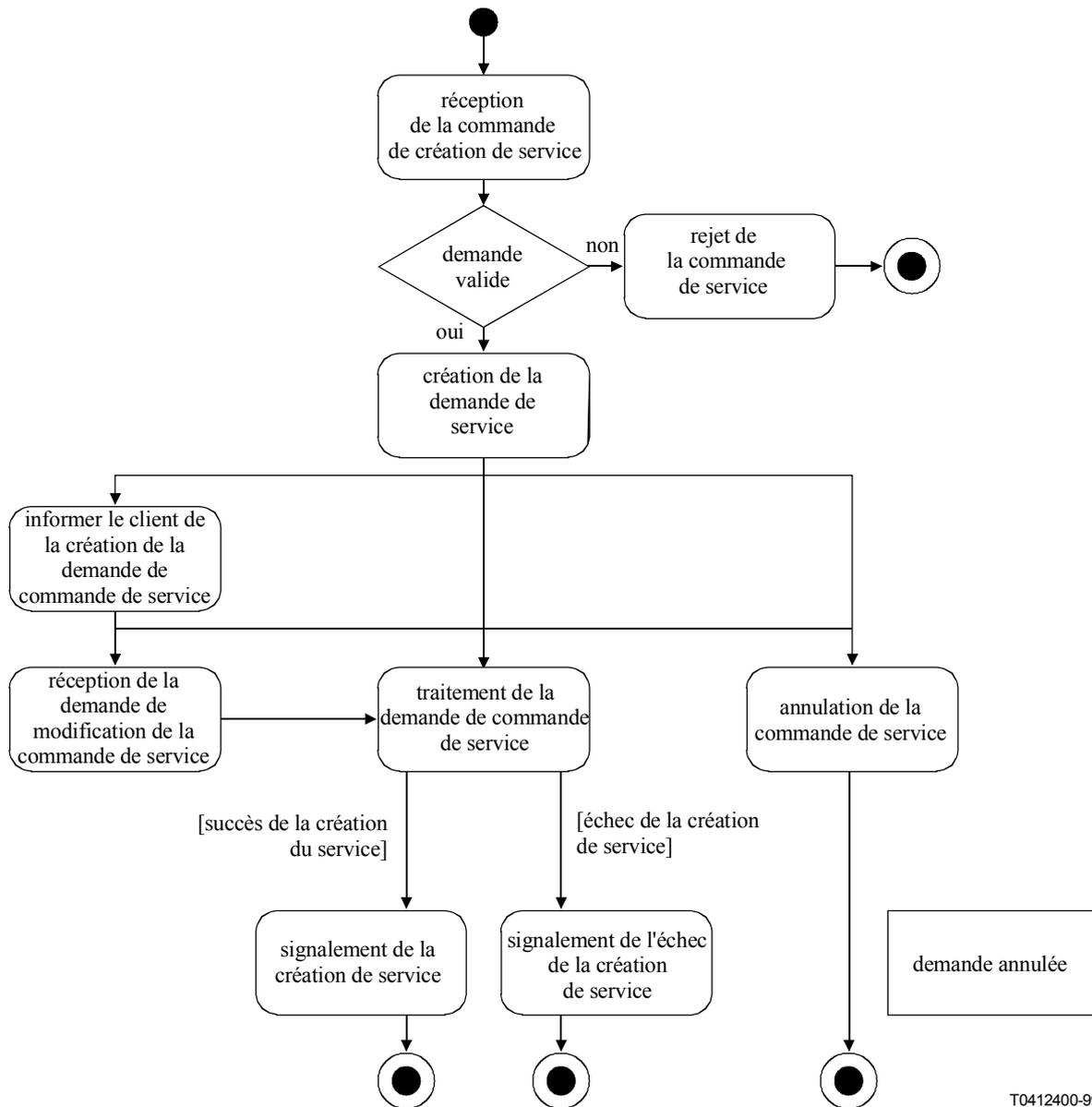
**Tableau I.1/M.3020 – Flux d'informations associé à la demande de création de service**

<b>Demande du client du service et réponse du SP</b>	<b>Client du service</b>	<b>Fournisseur de services</b>	<b>Notes</b>
Nom de service	m	o	Type de circuit loué offert par le SP. Les noms de service ne font pas l'objet de normalisation. Ils sont définis par contrat entre le SC et le SP.
Classe de service	o	c	Nom d'un profil de caractéristiques de service, associé au nom du service, défini et pris en charge par le SP. Exemples de caractéristiques de service que peut comporter une classe de service: directionnalité, partage de canal, options de signalisation, protection, objectifs de qualité de service, application, etc. Les noms de classes de service ne font pas l'objet de normalisation. Ils sont définis par le contrat. c – Si la classe de service demandée n'est pas égale à la classe de service qu'offre le SP, alors le SP doit fournir la valeur; sinon, la valeur est facultative.
Bande passante	o	c	Bande passante demandée, bande passante effective en retour. c – Si le SP ne peut pas offrir la bande passante demandée, le SP doit obligatoirement renvoyer la valeur accompagnée d'un code de raison indiquant que la bande passante demandée n'est pas disponible. Si la réponse n'indique pas l'aboutissement du traitement de la demande, le SP peut indiquer une condition d'erreur avec un code de raison indiquant que le service disponible diffère de la demande initiale de service du client.
Quantité	o	c	Nombre de circuits loués que le SP doit construire. Après traitement de la fonction de LCS, le SP doit obligatoirement renvoyer pour chaque circuit créé par le traitement un numéro de circuit unique.

### **I.2.2.3.5 Diagrammes d'activité**

Pour expliquer le déroulement des tâches lors de la création d'un nouveau service, on utilise un diagramme d'activité comme le montre la Figure I.8. Le diagramme d'activité montre les points où la synchronisation peut avoir lieu entre plusieurs activités. Dans cet exemple, le déroulement des tâches étant la représentation des activités chez un fournisseur de services, l'utilisation d'une notation telle que des couloirs (définissant des activités parallèles correspondant à plusieurs objets) n'est pas requise.

Le fournisseur de services reçoit la demande, la valide et crée ensuite la demande de service si elle est acceptée. Le client est informé de la bonne création et la barre de synchronisation indique que le traitement de la demande peut commencer parallèlement avec l'information du client. Après seulement que le client ait reçu le rapport de la création de l'objet demande de service, les demandes de modification de valeurs de paramètres peuvent être émises. De même, on peut émettre également un ordre d'annulation car, comme cela est noté plus loin, pour annuler une demande de service, la référence à l'objet demande de service créée est nécessaire. Même si cela n'est pas montré dans la figure, une fois que le top de synchronisation est apparu, les modifications de la progression de la demande peuvent être également signalées. Les activités qui résultent de la réussite ou de l'échec de la demande sont également représentées avec la fin du flux d'activité dans tous les cas.



T0412400-99

**Figure I.8/M.3020 – Diagramme d'activité: flux de tâches associé à la création d'un service**

NOTE – Cet exemple illustre l'utilisation de certaines notations de modélisation visuelle venant du langage UML pour décrire le comportement et les activités du service de gestion. Les classes d'objets représentées ici peuvent avoir une relation m:n avec celles dans la phase de conception (par exemple la classe d'objets gérés avec le paradigme CMIP et l'interface dans le CORBA/IDL). Ces figures peuvent être référencées dans la conception pour expliquer le comportement des entités propres au protocole.

### I.2.3 Conception

Les définitions GDMO pour cet exemple sont données dans la Recommandation UIT-T M.3108.1. Le comportement des classes d'objets gérés peut renvoyer au diagramme d'état présenté dans l'analyse.

Dans le contexte du GDMO, la hiérarchie et les diagrammes de nommage de la classe d'objets gérés exprimée en langage UML sont représentés ci-dessous. La traçabilité des divers éléments des modèles GDMO est présentée dans la Recommandation UIT-T M.3108.1.

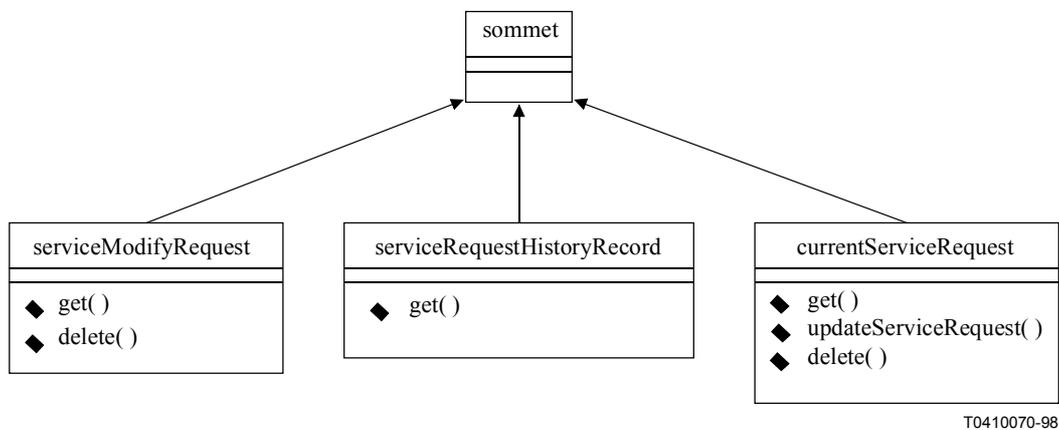
### I.2.3.1 Diagrammes de classes UML pour l'héritage des classes d'objets M.3208.1

Dans ces diagrammes, les classes sont représentées sous forme de rectangle comprenant trois parties: une dans la partie supérieure le nom de classes d'objets; dans la partie médiane, les noms d'attributs (non remplis dans ces figures pour des raisons de lisibilité), et dans la partie inférieure, les opérations d'accès.

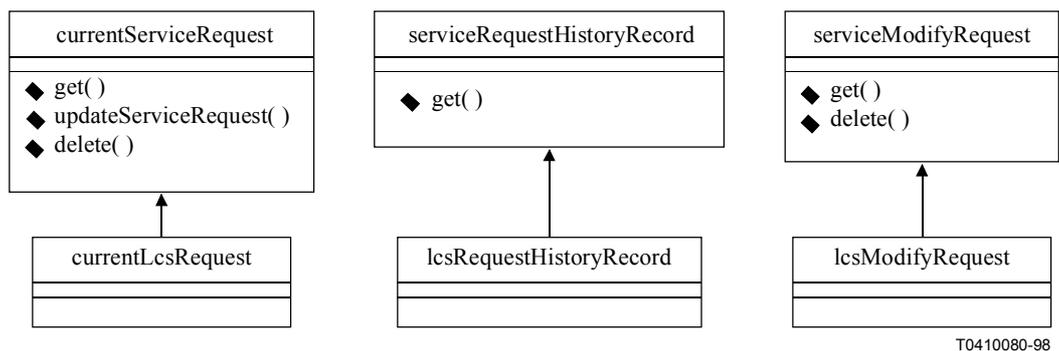
L'opération "get()" est utilisée dans les diagrammes de classes pour indiquer que les attributs de classe sont lisibles après la création d'une instance de cette classe.

L'opération "set()" est utilisée pour indiquer que certains (au moins un) attributs de classe peuvent être modifiés après la création d'une instance de cette classe.

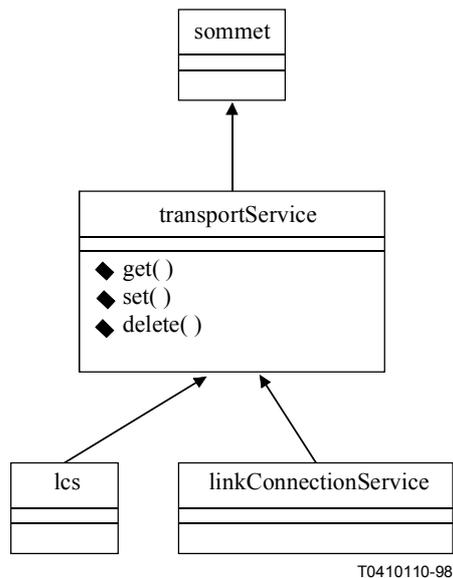
Les diagrammes de classe UML utilisent des flèches spéciales pour indiquer les relations d'héritage. Lorsqu'une classe est liée à une autre par héritage, les opérations provenant de la super classe (celle de la pointe de flèche) sont également prises en charge pour la classe héritée mais ne sont pas répétées dans la section opération du rectangle de classe. Voir Figures I.9-1 à I.9-3.



**Figure I.9-1/M.3020 – Relations d'héritage pour un fragment de demande de service générique**



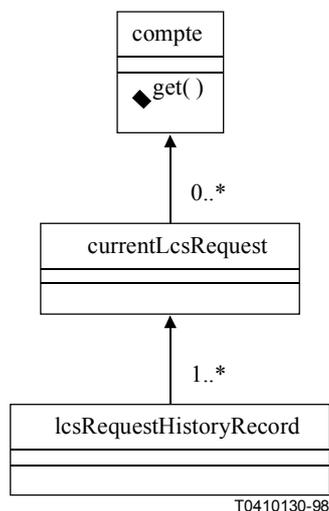
**Figure I.9-2/M.3020 – Relations d'héritage pour le fragment de demande de LCS**



**Figure I.9-3/M.3020 – Relations d'héritage pour le fragment de service**

### 1.2.3.2 Diagrammes de classe UML pour relations de modélisation

Les relations possibles entre des instances sont présentées dans les diagrammes de classe UML par des associations. Les relations de confinement sont indiquées par un segment dont le sommet est un losange qui touche le parent (agrégation UML). Des associations simples sont représentées par des droites dont les rôles sont indiqués à la fin de ces droites. Les cardinalités de relations sont indiquées par des étiquettes "0 .. \*" ou "1 .. \*" situées à la fin des droites représentant une relation. Les objets compte représentent l'acteur client du service présenté dans la section analyse. Voir Figure I.9-4.

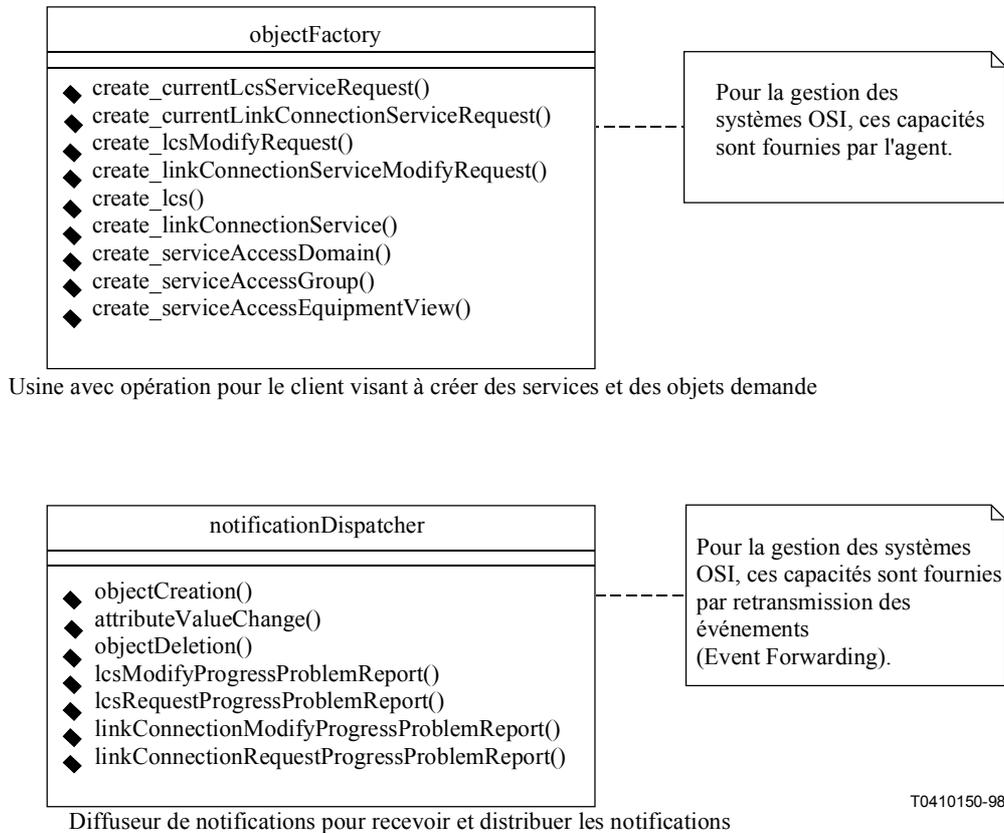


**Figure I.9-4/M.3020 – Relations de confinement pour la demande de service LCS avec relevé chronologique**

### 1.2.3.3 Diagrammes de classe UML pour la modélisation de la fonctionnalité agent

Certaines classes UML sont introduites (usines et diffuseur de notifications) pour modéliser les actions de création des objets et de distribution des notifications à partir des objets. Des instances de ces classes de fonctionnalité d'agent apparaissent dans les diagrammes séquentiels. Lorsqu'une

invocation d'opération de notification est formulée dans un objet de diffusion de notification, toutes les destinations intéressées recevront une copie de cette notification. Ces flux finaux de remise ne sont pas représentés dans les diagrammes à la Figure I.9-5, étant donné que de nombreux objets peuvent être intéressés de les recevoir.



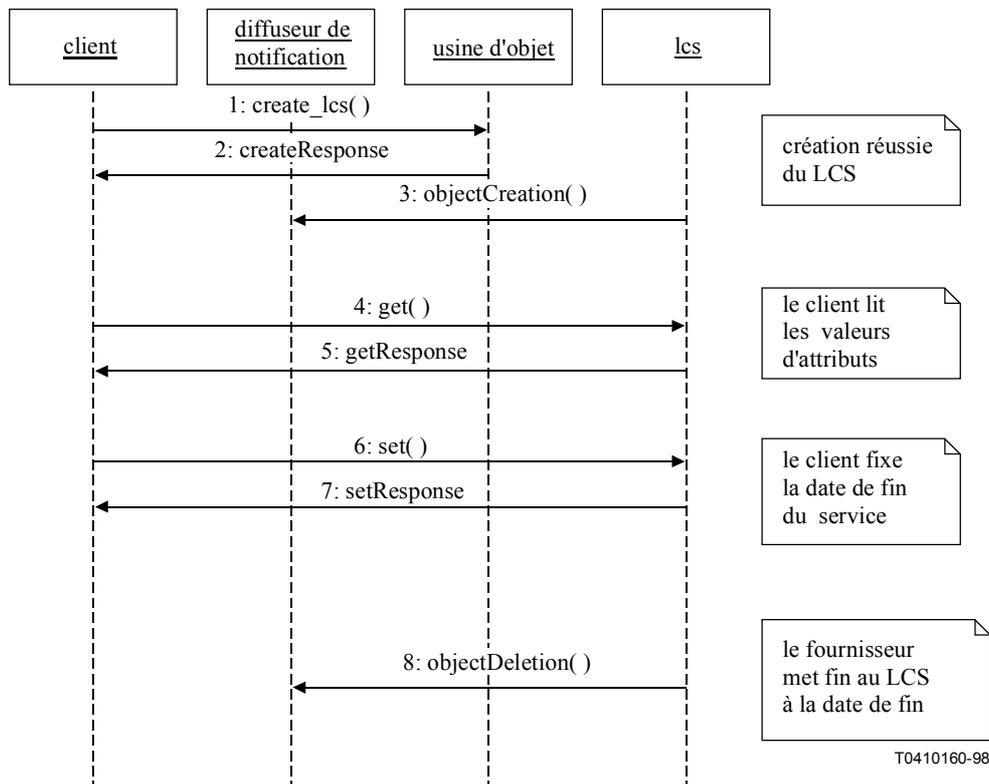
**Figure I.9-5/M.3020 – Modèles UML pour les fonctions d'agent (créant des objets et diffusant des notifications)**

#### I.2.3.4 Diagrammes séquentiels UML pour illustrer des scénarios d'utilisation d'objet

Les Figures I.9-1, I.9-2 et I.9-3 montrent l'utilisation de diagrammes séquentiels UML pour décrire des scénarios.

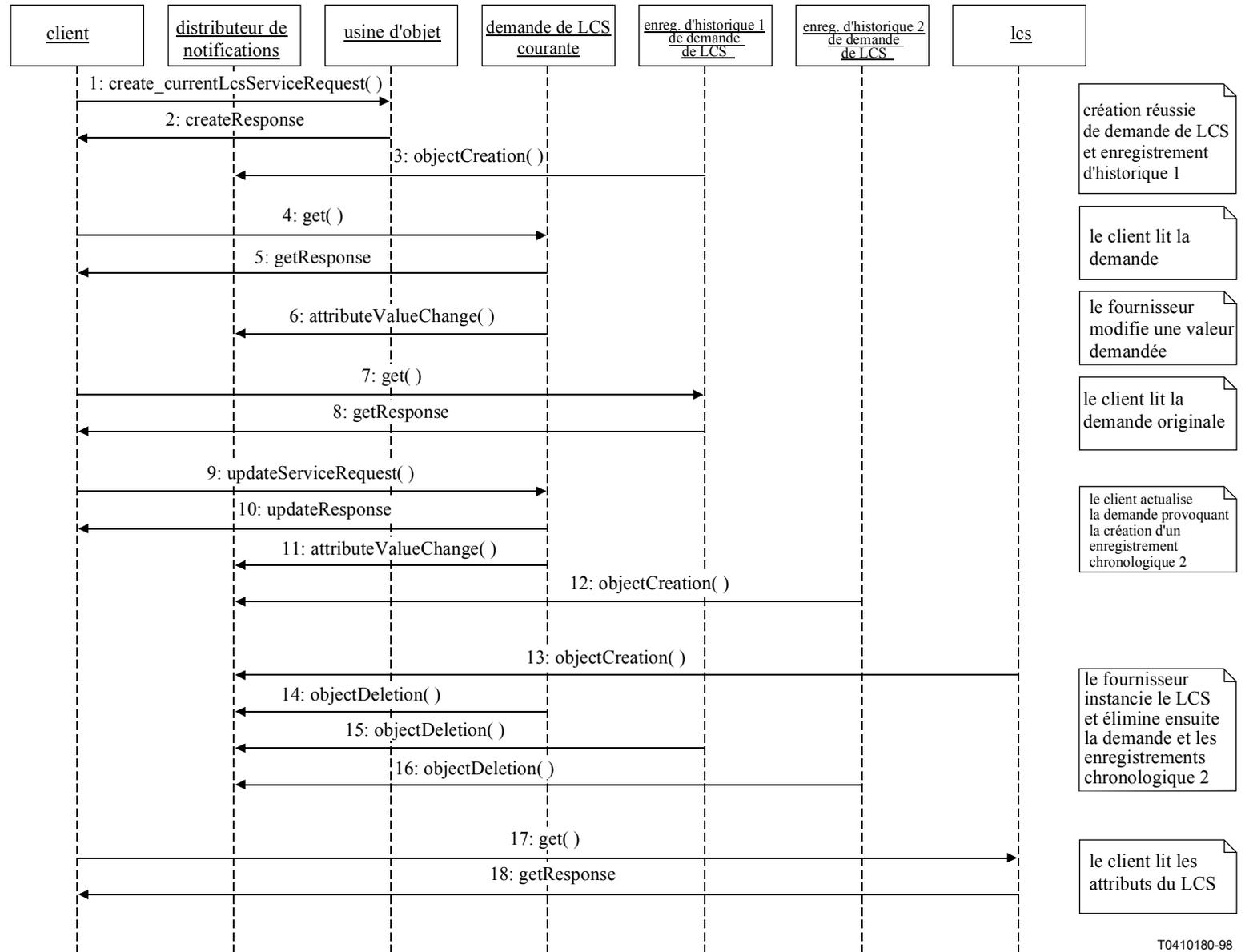
Les flux de messages depuis le diffuseur de notifications jusqu'aux destinations ultimes enregistrées ne sont pas représentés dans ces diagrammes. Il serait normal pour le client (ainsi que pour les autres objets) d'être un destinataire enregistré pour les notifications représentées dans ces diagrammes séquentiels.

La Figure I.9-6 illustre les échanges de messages pour créer une terminaison automatique du service LCS. Le flux dans ce diagramme utilise des classes d'objets représentant la fonctionnalité d'agent. Les étapes 1-3 peuvent être développées en étapes représentées à la Figure I.9-6. Les étapes intermédiaires qui sont autorisées (modifiant les valeurs de paramètre du service demandé) peuvent être notées en référant la figure dans la section d'analyse. La section d'analyse explique les critères pour la création réussie du service demandé et d'information du client. L'étape 3 à la Figure I.9-6 montre le nom de la notification applicable à cette conception. Un autre nom de message peut être utilisé dans une autre conception.



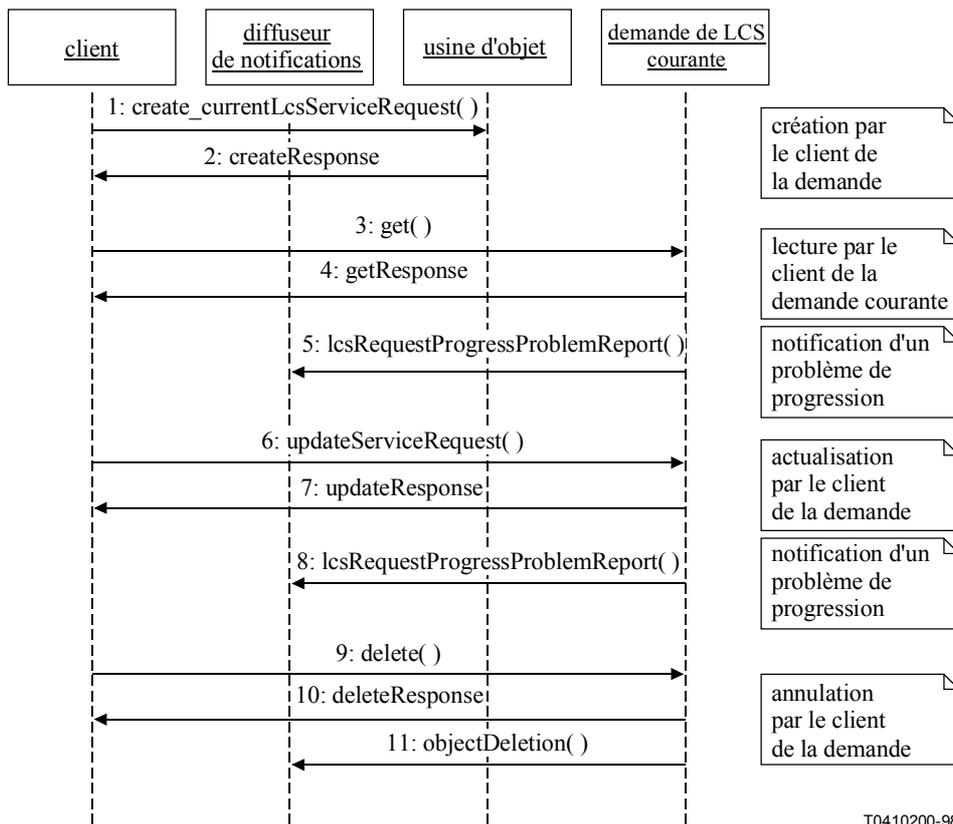
**Figure I.9-6/M.3020 – Diagramme séquentiel pour la création d'un service LCS et sa fin automatique explicite**

La Figure I.9-7 développe la Figure I.9-6 dans l'analyse avec des classes d'objets gérés spécifique définies dans le modèle GDMO. L'objet demande de service de la Figure I.9-6 est réalisé dans la conception en utilisant les objets demande de service LCS actuels et chronologiques. Le diagramme séquentiel dans ce cas de conception montre comment la chronologie de l'objet est créée, suite ou mise à jour de la demande originale. Le modèle dans la phase de conception affine encore plus l'objet demande de service dans la phase d'analyse de sorte que la chronologie des modifications demandées peut être conservée. Cela répond à l'exigence facultative identifiée au I.2.2.3.2 comme relevant du SLA. La Figure I.9-8 illustre le cas où la chronologie des changements ne peut pas être conservée. En outre, elle montre l'interaction entre les objets gérés dans la phase de conception lorsque la création du service est un échec (à noter que la figure correspondante pour l'échec de création n'est pas représentée dans la section analyse).



T0410180-98

**Figure I.9-7/M.3020 – Diagramme séquentiel d'instanciation de service LCS utilisant une demande et des enregistrements chronologiques**



**Figure I.9-8/M.3020 – Diagramme séquentiel d'échec d'instanciation de LCS utilisant la demande sans relevé chronologique**

## SÉRIES DES RECOMMANDATIONS UIT-T

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