

Union internationale des télécommunications

# UIT-T

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

# Q.818

(05/2012)

SÉRIE Q: COMMUTATION ET SIGNALISATION ET  
MESURES ET TESTS ASSOCIÉS

Interface Q3

---

**Services de gestion fondés sur des services  
web**

Recommandation UIT-T Q.818

UIT-T



RECOMMANDATIONS UIT-T DE LA SÉRIE Q  
COMMUTATION ET SIGNALISATION ET MESURES ET TESTS ASSOCIÉS

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4, 5, 6, R1 ET R2	Q.120–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.799
<b>INTERFACE Q3</b>	<b>Q.800–Q.849</b>
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRÉSCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
SPÉCIFICATIONS DE LA SIGNALISATION RELATIVE À LA COMMANDE D'APPEL INDÉPENDANTE DU SUPPORT	Q.1900–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999
SPÉCIFICATIONS ET PROTOCOLES DE SIGNALISATION POUR LES RÉSEAUX DE PROCHAINE GÉNÉRATION	Q.3000–Q.3709

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

# Recommandation UIT-T Q.818

## Services de gestion fondés sur des services web

### Résumé

La Recommandation UIT-T Q.818 définit un ensemble de services nécessaires pour la prise en charge des interfaces orientées service et fondées sur des services web et, conjointement avec la Recommandation UIT-T X.782, elle sert de cadre pour les interfaces de gestion de réseau fondées sur des services web. Elle établit des exigences relatives aux protocoles, indique comment utiliser certains services web bien connus dans des interfaces de gestion de réseau et définit certains services supports propres à la gestion de réseau. Elle fournit par ailleurs les définitions en langage WSDL des interfaces pour les services supports propres à la gestion de réseau.

### Historique

Edition	Recommandation	Approbation	Commission d'études	ID unique*
1.0	ITU-T Q.818	2012-05-14	2	<a href="http://handle.itu.int/11.1002/1000/11622">11.1002/1000/11622</a>

### Mots clés

Traitement réparti; langage de balisage extensible (XML); objets gérés; interfaces de gestion de réseau; orienté service; description; découverte et intégration universelles (UDDI); services web; langage de description des services web (WSDL); schéma XML.

---

\* Pour accéder à la Recommandation, reporter cet URL <http://handle.itu.int/> dans votre navigateur Web, suivi de l'identifiant unique, par exemple <http://handle.itu.int/11.1002/1000/11830-en>.

## AVANT-PROPOS

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

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

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

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

## NOTE

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

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

## 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 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 développeurs de consulter la base de données des brevets du TSB sous <http://www.itu.int/ITU-T/ipr/>.

© UIT 2018

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

## TABLE DES MATIÈRES

		<b>Page</b>
1	Domaine d'application .....	1
2	Références.....	1
3	Définitions .....	2
	3.1 Termes définis ailleurs .....	2
	3.2 Termes définis dans la présente Recommandation .....	2
4	Abréviations et acronymes .....	2
5	Conventions .....	3
6	Services de gestion de réseau fondés sur des services web: objectifs et exigences .....	3
	6.1 Objectifs .....	3
	6.2 Influence de la modélisation informationnelle .....	4
	6.3 Délimitation du domaine .....	5
	6.4 Notifications .....	6
7	Aperçu général du cadre et exigences relatives au protocole .....	6
	7.1 Aperçu général du cadre .....	6
	7.2 Exigences relatives au protocole, aux langages et aux services .....	7
8	Utilisation de services web bien connus .....	8
	8.1 Notification de base des services web .....	8
	8.2 Enregistrement de service UDDI.....	18
9	Services supports du cadre.....	23
	9.1 Service de pulsation.....	23
	9.2 Service d'opérations sur objets multiples .....	25
	9.3 Service de contenance .....	31
10	Observance et conformité .....	33
	10.1 Points de conformité du système .....	33
	Annexe A – Définition en WSDL des services supports du cadre .....	35
	A.1 Définition en WSDL et sous forme de schéma XML du service de notification de l'UIT .....	35
	A.2 Définition en WSDL et sous forme de schéma XML du service de pulsation de l'UIT .....	46
	A.3 Définition en WSDL et sous forme de schéma XML du service d'opérations MOO de l'UIT .....	48
	A.4 Définition en WSDL et sous forme de schéma XML du service de contenance de l'UIT .....	53
	Bibliographie.....	57



# Recommandation UIT-T Q.818

## Services de gestion fondés sur des services web

### 1 Domaine d'application

L'architecture de gestion de réseau définie dans la Recommandation [UIT-T M.3010] permet d'utiliser plusieurs protocoles de gestion. Jusqu'à présent, au niveau de la couche application, on pouvait utiliser la paire GDMO/CMIP et les protocoles GIOP/IOP de l'architecture CORBA. Grâce à la méthode de spécification des interfaces de gestion définie dans la Recommandation [UIT-T M.3020], il est possible de mettre en place de nouvelles technologies dans les interfaces de gestion de réseau et on peut désormais recourir aux services web/langage XML pour la gestion de réseau.

La présente Recommandation et la Recommandation [UIT-T Q.782] visent à déterminer un cadre permettant de définir la façon dont il convient de modéliser les interfaces prises en charge par les systèmes de gestion et par les éléments de réseau, au moyen des services web/schéma XML. La présente Recommandation a pour objet de fournir des lignes directrices ou des instructions concernant les points suivants:

- exigences relatives aux protocoles pour l'utilisation de services web pour la gestion de réseau;
- utilisation d'un service de notification fondé sur des services web dans les interfaces de gestion de réseau;
- enregistrement d'un nouveau service et accès à ce service au moyen de l'UDDI;
- contrôle de la disponibilité du mécanisme de transfert de notification des services web;
- accès à plusieurs objets gérés en une opération;
- exigences en matière d'observance et de conformité.

### 2 Références

Les Recommandations UIT-T et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions de la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Les Recommandations et autres références étant sujettes à révision, les utilisateurs de la présente Recommandation sont invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références énumérées ci-dessous. Une liste des Recommandations UIT-T en vigueur est publiée périodiquement. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

- [UIT-T X.701]      Recommandation UIT-T X.701 (1997), *Technologies de l'information – Interconnexion des systèmes ouverts – Aperçu général de la gestion-systèmes.*
- [UIT-T X.703]      Recommandation UIT-T X.703 (1997), *Technologies de l'information – Architecture de gestion répartie ouverte.*
- [UIT-T X.782]      Recommandation UIT-T X.782 (2012), *Lignes directrices relatives à la définition de services web pour les objets gérés et les interfaces de gestion.*
- [UIT-T M.3010]     Recommandation UIT-T M.3010 (2000), *Principes des réseaux de gestion des télécommunications.*
- [UIT-T M.3020]     Recommandation UIT-T M.3020 (2010), *Méthodologie pour la spécification des interfaces de gestion.*

[UIT-T M.3702]	Recommandation UIT-T M.3702 (2010), <i>Services communs de gestion – Gestion des notifications – Spécifications et analyse – Indépendance vis-à-vis du protocole</i>
[OASIS UDDI]	OASIS Specification (2004), <i>Universal Description, Discovery and Integration (UDDI) v3.0.2.</i>
[OASIS WSN]	OASIS Specification (2006), <i>Web Services Base Notification v1.3.</i>
[W3C SOAP]	W3C Recommendation (2007), <i>SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).</i>
[W3C XML]	W3C Recommendation (2000), <i>Extensible Markup Language (XML) 1.0 Second Edition.</i>
[W3C XS-P1]	W3C Recommendation (2004), <i>XML Schema Part 1: Structures Second Edition.</i>
[W3C XS-P2]	W3C Recommendation (2004), <i>XML Schema Part 2: Datatypes Second Edition.</i>
[W3C WSDL]	W3C Recommendation (2001), <i>Web Services Description Language (WSDL) 1.1.</i>

### 3 Définitions

#### 3.1 Termes définis ailleurs

La présente Recommandation utilise les termes suivants définis ailleurs:

- 3.1.1 **agent** [UIT-T M.3020]
- 3.1.2 **classe d'objets gérés** [UIT-T X.701]
- 3.1.3 **gestionnaire** [UIT-T M.3020]
- 3.1.4 **notification** [UIT-T X.703]

#### 3.2 Termes définis dans la présente Recommandation

Néant.

### 4 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

CMIP	protocole commun d'informations de gestion ( <i>common management information protocol</i> )
CORBA	architecture de courtier commun de requête sur des objets ( <i>common object request and broker architecture</i> )
GDMO	directives pour la définition des objets gérés ( <i>guidelines for the definition of managed objects</i> )
GIOP	protocole général entre courtiers ORB ( <i>general inter-ORB protocol</i> )
IIOB	protocole Internet entre courtiers ORB ( <i>internet inter-ORB protocol</i> )
MO	objet géré ( <i>managed object</i> )
MOC	classe d'objets gérés ( <i>managed object class</i> )
MOO	opérations sur objets multiples ( <i>multiple object operation</i> )
RGT	réseau de gestion des télécommunications
SOAP	protocole simple d'accès aux objets ( <i>simple object access protocol</i> )

UDDI	description, découverte et intégration universelles ( <i>universal description discovery and integration</i> )
WS	services web ( <i>Web services</i> )
WSDL	langage de description des services web ( <i>Web services description language</i> )
WSN	notification des services web ( <i>Web services notification</i> )
XML	langage de balisage extensible ( <i>extensible markup language</i> )
XSD	définition de schéma XML ( <i>XML schema definition</i> )

## 5 Conventions

Un petit nombre de conventions sont appliquées dans la présente Recommandation afin d'informer le lecteur de l'intention du texte. Bien que la plus grande partie de la Recommandation soit normative, les alinéas qui décrivent succinctement les exigences obligatoires qu'un système de gestion (gérant ou géré) doit respecter sont précédés de la lettre "R" en caractère gras entre parenthèses, suivie d'un nom court indiquant l'objet de l'exigence et d'un numéro. Par exemple:

**(R) EXEMPLE-1** Exemple d'exigence obligatoire.

Les exigences dont l'application par un système de gestion est facultative sont précédées de la lettre "O" au lieu de "R". Par exemple:

**(O) OPTION-1** Exemple d'exigence facultative.

Les déclarations d'exigences servent à créer des profils d'observance et de conformité.

On trouvera dans la présente Recommandation de nombreux exemples en WSDL et sous forme de schéma XML; des exemples en WSDL de services propres à la gestion et de types de données supports figurent dans une annexe normative. Les exemples en WSDL et les schémas XML sont écrits en caractères Courier New de taille 10:

```
<!-- Example WSDL -->
<wsdl:message name="exampleMessage">
  <wsdl:part name="exampleRequest" type="exampleRequestType"/>
</wsdl:message>
```

## 6 Services de gestion de réseau fondés sur des services web: objectifs et exigences

Le présent paragraphe décrit les principaux objectifs du cadre des services ainsi que les exigences facilitant la prise en charge de ces objectifs par les services de gestion de réseau fondés sur des services web. Le § 6.1 présente les objectifs du cadre des services web. Le § 6.2 contient la terminologie et les exigences. Les exigences du présent paragraphe sont celles que le cadre général doit observer et sont fondées sur des besoins en gestion des télécommunications. Les § 7, 8 et 9 décrivent ensuite un cadre répondant à ces besoins et définissant la façon de satisfaire aux exigences du présent paragraphe en utilisant d'une certaine façon les services web.

### 6.1 Objectifs

La présente Recommandation vise à déterminer un cadre permettant de définir la façon dont il convient de modéliser les interfaces prises en charge par les systèmes de gestion et par les éléments de réseau. Les objectifs essentiels du cadre sont notamment les suivants:

- interopérabilité des applications;
- usage commun de services web bien connus.

Le présent paragraphe développe ces deux objectifs.

### **6.1.1 Interopérabilité des applications**

Un des objectifs essentiels de l'architecture de gestion de réseau (et en particulier de l'architecture d'information), consiste à promouvoir un cadre normalisé pour assurer l'interopérabilité et l'échange d'informations entre systèmes à partir d'un ensemble de divers fournisseurs de systèmes de gestion de réseau. L'interopérabilité entre systèmes implique de nombreux aspects de développement. Au niveau de la couche la plus basse, un mécanisme commun de communication doit être en place afin de prendre en charge une syntaxe commune, l'établissement de la connectivité et l'échange entre systèmes des demandes/réponses des opérations. Cet aspect de l'interopérabilité est implicitement pris en charge par les spécifications des services web.

Pour la gestion de réseau, il est nécessaire d'assurer l'interopérabilité des applications. En d'autres termes, des systèmes de gestion issus de divers fournisseurs seront utilisés dans un même RGT d'administration afin de prendre en charge différentes fonctions nécessaires pour assurer la gestion de ses réseaux. Afin de simplifier l'intégration des systèmes de ces divers fournisseurs, ceux-ci doivent s'entendre sur la sémantique des informations à échanger. C'est à cette fin qu'un modèle d'information est spécifié. Les directives pour la définition de modèles d'information fondés sur des services web sont spécifiées dans la Recommandation [UIT-T X.782], mais les services définis ci-après devraient être compatibles avec ces directives.

### **6.1.2 Usage commun de services web bien connus**

Un deuxième aspect de ce cadre général est la définition de l'usage commun et la configuration de l'environnement de traitement réparti choisi. Cet aspect du cadre devra indiquer les anticipations raisonnables que les fournisseurs de système de gestion de réseau peuvent avoir les uns envers les autres. Plutôt que de redéfinir les capacités d'interface requises afin de prendre en charge des fonctions de gestion de réseau communes à chaque modèle d'information, comme le filtrage des notifications, les lignes directrices de modélisation de la Recommandation [UIT-T X.782] se fondent sur un ensemble de services supports qui permettent de simplifier les modèles d'information tout en améliorant l'interopérabilité.

Lors de la définition de ces services, un effort spécial sera consenti pour faire usage de certains services web bien connus. Plus précisément, la présente Recommandation traitera de l'utilisation de la notification des services web, telle que définie dans [OASIS WSN], et de l'enregistrement de service UDDI pour les services web, tel que défini dans [OASIS UDDI], qui auront une incidence sur l'interopérabilité des systèmes (c'est-à-dire que les questions impliquant l'utilisation de services web dans un seul système sont hors du domaine d'application de la présente Recommandation). Lorsque les besoins de la gestion de réseau ne pourront pas être satisfaits par les services web bien connus susmentionnés, des services additionnels seront définis.

## **6.2 Influence de la modélisation informationnelle**

Comme décrit dans le paragraphe précédent, la modélisation explicite des ressources gérables de part et d'autre d'une interface est au centre de l'interopérabilité des applications. Les lignes directrices relatives à la définition des objets gérés fondés sur des services web détaillées dans la Recommandation [UIT-T X.782] décrivent les règles de modélisation des ressources gérables. Elles impliquent également plusieurs décisions qui doivent être prises en charge par le cadre des services de gestion de réseau fondés sur des services web. Le présent paragraphe résume ces points.

### **6.2.1 Granularité d'accès**

La granularité d'interface des services web se rapporte à la relation entre les ressources qui sont modélisées à une interface donnée et les moyens par lesquels on y accède par l'intermédiaire de services web. La Recommandation [UIT-T X.782] utilise une approche de modélisation à granularité de service, ce qui signifie que chaque ressource modélisée n'est accessible qu'au moyen d'un service web particulier. Les objets qui représentent des ressources gérables sont appelés objets gérés.

## 6.2.2 Représentation des relations de contenance et de nommage

La contenance est une représentation logique de la façon dont des ressources modélisées contiennent d'autres ressources modélisées. Traditionnellement, la contenance est une relation très importante dans les applications de gestion de réseau car c'est un moyen pratique d'identification du grand nombre de ressources qui doivent être gérées. Selon les lignes directrices de la Recommandation [UIT-T X.782], un nom unique doit être attribué à chaque objet géré, fondé en partie sur le nom de l'objet qui le contient. Les services de gestion fondés sur des services web doivent toujours permettre de mémoriser ces noms (et donc les relations de contenance qu'ils représentent).

## 6.2.3 Création et suppression d'objets

Les plates-formes de services web ne permettent pas aux clients de créer ou de supprimer des objets dans des systèmes distants. Par contre, ces fonctionnalités sont fournies par les systèmes distants, qui permettent aux clients de créer ou de supprimer des objets dans le système distant. La Recommandation [UIT-T X.782] fournit un service commun d'accès aux objets, de sorte que la création et la suppression d'objets soient indépendantes du modèle. Lors de la suppression d'un objet contenant d'autres objets, il convient de conserver la relation de contenance, afin qu'il n'y ait aucun objet orphelin dans les systèmes distants. L'information relative à la contenance devrait être conservée dans le système distant et tenue à jour lors des opérations de création et de suppression.

## 6.3 Délimitation du domaine

La possibilité d'effectuer des requêtes complexes (c'est-à-dire d'opérations GET), des mises à jour (c'est-à-dire d'opérations SET) et des opérations de suppression sur un groupe d'entités au moyen d'une unique demande d'opération est une précieuse caractéristique du RGT. Les systèmes de gestion peuvent avoir à gérer jusqu'à  $10^7$  instances d'objets gérés. Compte tenu des dimensions de la base d'informations de gestion, un système gérant ne peut pas exécuter efficacement des requêtes ad hoc visant des instances individuelles d'objets gérés (c'est-à-dire d'entités). Le système gérant s'attend plutôt à ce que le système géré prenne en charge un certain niveau d'intelligence.

L'intelligence contenue dans le système géré permet au système gérant de sélectionner un groupe d'entités gérées sur lesquelles une opération sera exécutée. La sélection des entités gérées repose sur la délimitation d'un domaine. Ce processus de sélection d'entités gérées est assuré par un service qui sera défini plus loin et qui permet à un système gérant de sélectionner un domaine d'objets sur lesquels il pourra agir (ce domaine étant défini au moyen de relations de contenance; voir le § 6.2.2). Une fois que le domaine des entités est déterminé, l'opération (spécifiée par la demande limitée au domaine) est exécutée seulement sur ces entités.

L'utilisation de la délimitation du domaine dans le présent cadre général permet d'effectuer les opérations suivantes:

- obtention limitée à un domaine: cette opération retourne les valeurs issues de chacune des entités qui relèvent du domaine (pour une liste d'attributs);
- mise à jour limitée à un domaine: cette opération remplace une valeur d'attribut ou ajoute/retranche des valeurs à des attributs à valeur d'ensemble, dans le groupe des entités qui relèvent du domaine, en fonction des valeurs spécifiées dans la demande limitée au domaine. Cette opération peut servir à mettre à jour un ou plusieurs attributs contenus dans un même objet ou dans plusieurs objets;
- suppression limitée à un domaine: cette opération supprime toutes les entités qui relèvent du domaine.

La délimitation d'un domaine implique l'identification des entités auxquelles un filtre doit être appliqué. Elle est effectuée sur la base de la hiérarchie de contenance définie au § 6.2.2. La délimitation du domaine est appliquée à partir d'une entité gérée de base en descendant dans l'arbre de contenance jusqu'à une profondeur déterminée.

L'entité de base du domaine est définie comme étant la racine de l'arbre de contenance à partir de laquelle la recherche doit commencer. Une demande limitée à un domaine doit toujours spécifier l'entité gérée de base du domaine. La profondeur du niveau de la délimitation du domaine peut ensuite être spécifiée de l'une des quatre façons suivantes à l'intérieur de la demande en question:

- 1) selon l'entité de base;
- 2) selon les subordonnés du n<sup>ième</sup> niveau de l'entité de base;
- 3) selon l'entité de base et tous ses subordonnés jusqu'au n<sup>ième</sup> niveau inclus;
- 4) selon l'entité de base et tous ses subordonnés (c'est-à-dire l'ensemble du sous-arbre).

## **6.4 Notifications**

Le cadre doit pouvoir prendre en charge les capacités suivantes:

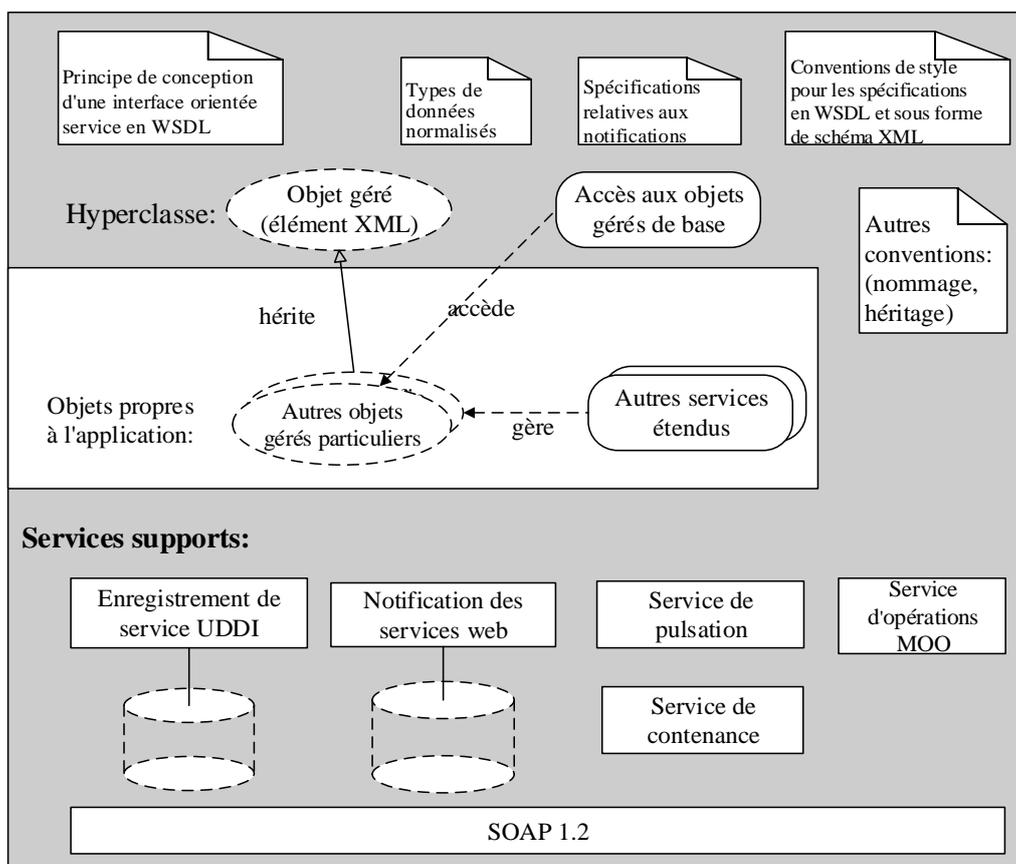
- remise de notifications;
- abonnement à des types de notification;
- renvoi de notifications à des destinations multiples;
- filtrage des notifications;
- identification univoque de la ressource émettant la notification.

## **7 Aperçu général du cadre et exigences relatives au protocole**

### **7.1 Aperçu général du cadre**

Ce cadre relatif aux interfaces de gestion fondées sur des services web est un ensemble de capacités dont un élément est un ensemble de services web bien connus. Il définit le rôle de ces services dans les interfaces de gestion de réseau ainsi que les conventions relatives à leur utilisation. Le cadre définit également des services supports qui n'ont pas encore été normalisés dans le secteur privé mais qui sont appelés à être normalisés dans les interfaces de gestion de réseau conformes à ce cadre. Les interfaces pour ces services sont définies en WSDL dans l'Annexe A.

Pour prendre en charge les objets logiciels représentant des ressources gérables, le cadre exige que ces objets mettent en oeuvre certaines capacités communes de base. Une classe de base est donc définie dans la Recommandation [UIT-T X.782] pour usage lors de la modélisation de ressources de gestion de réseau. Les classes d'objets gérés contenues dans les modèles d'information doivent toujours hériter d'un ensemble de base de capacités issues de la classe de base et pouvoir les réaliser afin de fonctionner à l'intérieur de ce cadre. Enfin, certaines règles et conventions sont définies pour les développeurs de modèles d'information à utiliser avec ce cadre. Elles consistent en directives de modélisation et en idiomes de type XML. Toutes ces fonctions sont décrites graphiquement dans la Figure 1.



**Figure 1 – Aperçu général du cadre**

Dans la Figure 1, le cadre est représenté en gris. Au centre se trouvent les objets propres à une application qui sont pris en charge par le cadre. Dans la rangée du bas, le cadre contient le protocole de communication: SOAP. Au-dessus, un certain nombre de cases contenant des noms représentent les services qui composent le cadre (certaines cases sont reliées à des icônes décrivant les bases de données qu'elles devront tenir à jour afin d'exécuter leurs fonctions). En haut de la figure, des icônes représentent l'hyperclasse des objets gérés ainsi que le service d'accès aux objets gérés de base. Chacun des objets gérés pris en charge par ce cadre doit finalement hériter de l'hyperclasse. La Figure 1 montre également des icônes de pages aux coins retournés, qui représentent des conventions normalisées de modélisation d'objet.

Les services du cadre qui sont représentés sous la forme de cases à coins quadrangulaires sont définis dans la présente Recommandation. L'hyperclasse, les notifications et les conventions de modélisation d'objets sont définies dans la Recommandation [UIT-T X.782].

## 7.2 Exigences relatives au protocole, aux langages et aux services

Ce paragraphe définit les versions requises des langages, du protocole et des services pour prendre en charge le cadre. Les spécifications relatives à la technologie de base et au protocole des services web sont définies par le W3C. De plus, deux services bien connus utilisés dans le cadre sont définis par l'OASIS. Le Tableau 1 montre quelle est la version (y compris les versions ultérieures publiées jusqu'à la date de la présente Recommandation) des spécifications applicables qui doit être prise en charge afin d'assurer la conformité au cadre.

**Tableau 1 – Versions du langage, du protocole et des services des services web**

Service	Version
Recommandation sur le langage WSDL (W3C)	1.1 (mars 2001)
Recommandation sur les schémas XML (W3C)	1.1 (octobre 2004)
Notification de base des services web ([OASIS WSN])	1.3 (octobre 2006)
Spécification UDDI ([OASIS UDDI])	3.0.2 (octobre 2004)
SOAP (W3C)	1.2 (avril 2007)

## 8 Utilisation de services web bien connus

### 8.1 Notification de base des services web

Le cadre décrit fait appel à la spécification OASIS relative à la notification de base des services web pour le transfert des notifications demandées dans les interfaces de gestion de réseau.

#### 8.1.1 Aperçu général de la notification de base des services web

La notification de base des services web (WSN) est une spécification qui définit une approche normalisée pour les services web en matière de notification au moyen d'un modèle thématique de publication/abonnement. La spécification WSN comprend les échanges normalisés de messages que doivent mettre en oeuvre les fournisseurs de notifications, les exigences opérationnelles attendues des fournisseurs et demandeurs de services qui participent aux notifications, ainsi qu'un modèle XML décrivant les thèmes.

La notification de base des services web (WSN) assure l'échange asynchrone de messages d'événements entre clients au moyen d'un paradigme de type abonnement-publication, tel que décrit dans [OASIS WSN]. Elle permet aussi le filtrage des notifications qui sont requises par ailleurs dans les interfaces de gestion de réseau.

#### 8.1.2 Mappage des opérations entre la Recommandation UIT-T M.3702 et la spécification WSN

Les mécanismes et les interfaces en WSDL de la spécification [OASIS WSN] seront utilisés dans ce cadre. Cependant, ils seront considérés comme des fonctionnalités internes, qui ne seront pas mises directement à la disposition des destinataires des notifications. La Recommandation [UIT-T M.3702] établit des spécifications et des analyses indépendantes du protocole pour les services communs de gestion des notifications, qui définissent les fonctions communes devant être mises en oeuvre par un paradigme de gestion propre à un protocole. Dans le cadre décrit, le modèle commun de gestion des notifications est mappé sur les définitions en WSDL/XML, et certaines opérations en WSDL sont ensuite mappées sur des opérations de la spécification [OASIS WSN].

Le tableau suivant indique le mappage entre la Recommandation [UIT-T M.3702] et la notification des services web.

**Tableau 2 – Mappage entre la Recommandation [UIT-T M.3702], la présente Recommandation et la spécification [OASIS WSN]**

N°	[UIT-T M.3702]		UIT-T Q.818		[OASIS WSN]	
	Nom de l'opération		Nom de l'opération		Opération	
	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres
1	subscribeNotification		subscribeNotification		Subscribe	

**Tableau 2 – Mappage entre la Recommandation [UIT-T M.3702], la présente Recommandation et la spécification [OASIS WSN]**

N°	[UIT-T M.3702]		UIT-T Q.818		[OASIS WSN]	
	Nom de l'opération		Nom de l'opération		Opération	
	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres
	Paramètre d'entrée	<ul style="list-style-type: none"> <li>- managerId</li> <li>- notificationTypes</li> <li>- filteringCriteria</li> <li>- destination</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- managerId: type="nts:IdType"</li> <li>- notificationTypes: type="nts:Notification TypeListType"</li> <li>- filteringCriteria type="nts:FilterType"</li> <li>- destination: type= "wsa:EndpointReference Type"</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- consumerReference,</li> <li>- filter,</li> <li>- subscriptionPolicy,</li> <li>- initialTerminationTime</li> </ul>
	Paramètre de sortie	<ul style="list-style-type: none"> <li>- subscriptionId</li> <li>- status</li> </ul>	Réponse	<ul style="list-style-type: none"> <li>- subscriptionId: type="nts:IdType"</li> <li>- status: type="nts:StatusType"</li> </ul>	Réponse	subscriptionReference
2	unsubscribeNotification		unsubscribeNotification		Unsubscribe	
	Paramètre d'entrée	<ul style="list-style-type: none"> <li>- managerId</li> <li>- subscriptionId</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- managerId type="nts:IdType"</li> <li>- subscriptionId type="nts:IdType"</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- subscriptionReference</li> </ul>
	Paramètre de sortie	<ul style="list-style-type: none"> <li>- status</li> </ul>	Réponse	<ul style="list-style-type: none"> <li>- status: type="nts:StatusType"</li> </ul>	Réponse	NULL
3	suspendSubscription		suspendSubscription		PauseSubscription	
	Paramètre d'entrée	<ul style="list-style-type: none"> <li>- managerId</li> <li>- subscriptionId</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- managerId: type="nts:IdType"</li> <li>- subscriptionId: type="nts:IdType"</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- subscriptionReference</li> </ul>
	Paramètre de sortie	<ul style="list-style-type: none"> <li>- status</li> </ul>	Réponse	<ul style="list-style-type: none"> <li>- status type="nts:StatusType"</li> </ul>	Réponse	NULL
4	resumeSubscription		resumeSubscriptions		ResumeSubscription	
	Paramètre d'entrée	<ul style="list-style-type: none"> <li>- managerId</li> <li>- subscriptionId</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- managerId: type="nts:IdType"</li> <li>- subscriptionId: type="nts:IdType"</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- subscriptionReference</li> </ul>
	Paramètre de sortie	<ul style="list-style-type: none"> <li>- status</li> </ul>	Réponse	<ul style="list-style-type: none"> <li>- status: type="nts:StatusType"</li> </ul>	Réponse	NULL
5	getNotificationTypes		getNotificationTypes		--	
	Paramètre d'entrée	<ul style="list-style-type: none"> <li>- notificationIRPId</li> </ul>	Demande	<ul style="list-style-type: none"> <li>- notificationIRPId: type="x782:NameType"</li> </ul>		
	Paramètre de sortie	<ul style="list-style-type: none"> <li>- status</li> <li>- notificationType List</li> </ul>	Réponse	<ul style="list-style-type: none"> <li>- notificationTypeList: type="nts:Notification TypeListType"</li> <li>- status type="nts:StatusType"</li> </ul>		

**Tableau 2 – Mappage entre la Recommandation [UIT-T M.3702], la présente Recommandation et la spécification [OASIS WSN]**

N°	[UIT-T M.3702]		UIT-T Q.818		[OASIS WSN]	
	Nom de l'opération		Nom de l'opération		Opération	
	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres	Demande/ réponse	Paramètres
6	querySubscription		querySubscription		-	
	Paramètre d'entrée	- subscriptionId	Demande	- subscriptionId: type="nts:IdType"		
	Paramètre de sortie	- subscribed NotificationType - subscriptionStatus - destination - filteringCriteria - status	Réponse	- notificationTypes: type="nts:Notification TypeListType" - subscriptionStatus: type="nts:Subscription StatusType" - destination: type="wsa: EndpointReference Type" - filteringCriteria: type= "nts:FilterType" - status type="nts:StatusType"		
7	modifySubscription		modifySubscription		--	
	Paramètre d'entrée	- subscriptionId - filteringCriteria - destination - notificationTypes	Demande	- subscriptionId: type="nts:IdType" - filteringCriteria: type="nts:FilterType" - destination: type="wsa: EndpointReference Type" - notificationTypes: type="nts:Notification TypeListType"		
	Paramètre de sortie	- status	Réponse	- status type="nts:StatusType"		
8	listAllSubscriptionIds		listAllSubscriptionIds		--	
	Paramètre d'entrée	- managerId	Demande	- managerId: type="nts:IdType"		
	Paramètre de sortie	- subscriptionIdSet - status	Réponse	- subscriptionIdSet: type="nts:IdSetType" - status type="nts:StatusType"		
9	Notification de rapport:		- (Utilisation directe de l'opération <i>notify</i> WSN)		Notify:	
					Message de la notification	- SubscriptionReference - Topic - ProducerURL - Message
					Réponse	NULL

Le Tableau 3 fournit des descriptions complémentaires relatives au mappage entre les opérations définies dans la présente Recommandation et celles qui sont définies dans la notification WSN.

**Tableau 3 – Descriptions complémentaires relatives au mappage des opérations de la Recommandation [UIT-T M.3702]**

Nom de l'opération	Demande/ réponse	UIT-T Q.818	[OASIS WSN]
subscribe	Demande	<i>managerId</i>	Il n'y a pas de paramètre <i>managerId</i> dans la notification WSN. Il sera traité et stocké uniquement par un agent, et non par la notification WSN.
		<i>filterCriteria</i>	<i>Filter</i>
		<i>destination</i>	<i>ConsumerReference</i>
		<i>NotificationTypes</i>	Il n'y a pas de mappage direct de <i>NotificationTypes</i> avec la notification WSN, mais le paramètre <i>filter</i> de la notification WSN peut assurer la fonction des types de notifications qui prennent en charge le filtrage.
		–	<i>subscriptionPolicy</i> Ce paramètre est facultatif dans la notification WSN. Il peut contenir des données brutes ou une valeur quelconque dans le message <i>Notify</i> . Ce dernier peut fournir davantage d'informations dans un message, conformément à la définition de la notification WSN. Dans le mappage de ce type d'opérations, la valeur de <i>subscriptionPolicy</i> devrait toujours utiliser le message <i>Notify</i> de la présente Recommandation.
–	<i>initialTerminationTime</i> <i>initialTerminationTime</i> indique le moment où un abonnement prend fin. Lorsque la valeur de l'attribut <i>xsi:nil</i> est "vrai", il n'y a pas de limite à l'abonnement. Dans la présente Recommandation, la valeur par défaut pour ce paramètre de la notification WSN est <i>xsi:nil = true</i> (vrai).		
	Réponse	subscriptionId	subscriptionReference
unsubscribe	Demande	<i>managerId</i>	Le concept de <i>managerId</i> , qui sera traité et stocké dans l'agent lui-même, n'a pas d'équivalent dans la notification WSN. Le cas échéant, il convient d'utiliser une donnée de type chaîne.
		subscriptionId	subscriptionReference
suspendSubscription	Demande	<i>managerId</i>	Le concept de <i>managerId</i> , qui sera traité et stocké dans l'agent lui-même, n'a pas d'équivalent dans la notification WSN. Le cas échéant, il convient d'utiliser une donnée de type chaîne.
		subscriptionId	subscriptionReference
resumeSubscription	Demande	<i>managerId</i>	Le concept de <i>managerId</i> , qui sera traité et stocké dans l'agent lui-même, n'a pas d'équivalent dans la notification WSN. Le cas

**Tableau 3 – Descriptions complémentaires relatives au mappage des opérations de la Recommandation [UIT-T M.3702]**

Nom de l'opération	Demande/ réponse	UIT-T Q.818	[OASIS WSN]
			échéant, il convient d'utiliser une donnée de type chaîne.
		subscriptionId	subscriptionReference
getNotificationTypes		Il n'y a pas d'opérations correspondantes définies dans la notification WSN. Elles devraient être mises en oeuvre par un agent lui-même.	
querySubscription			
modifySubscription			
listAllSubscriptionIds			
notify		Dans la présente Recommandation, l'opération <i>notify</i> de la notification WSN est utilisée directement dans un agent.	

Une définition détaillée de l'interface figure au § A.1.

### 8.1.3 Définitions relatives au format de notification

#### 8.1.3.1 Format de notification à utiliser dans le cadre de la présente Recommandation

Le tableau suivant décrit les opérations *notify* définies dans [OASIS WSN] qui seront utilisées dans le cadre de la présente Recommandation pour l'envoi de notifications.

**Tableau 4 – Opérations *notify* provenant de [OASIS WSN]**

<pre> &lt;wsnt:Notify&gt;   &lt;wsnt:NotificationMessage&gt;     &lt;wsnt:SubscriptionReference&gt;       wsa:EndpointReferenceType     &lt;/wsnt:SubscriptionReference&gt; ?     &lt;wsnt:Topic Dialect="xsd:anyURI"&gt;       {any} ?     &lt;/wsnt:Topic&gt;?     &lt;wsnt:ProducerReference&gt;       wsa:EndpointReferenceType     &lt;/wsnt:ProducerReference&gt; ?     &lt;wsnt:Message&gt;       {any}     &lt;/wsnt:Message&gt;   &lt;/wsnt:NotificationMessage&gt; +   {any} * &lt;/wsnt:Notify&gt; </pre>
--

Dans la présente Recommandation, tous les composants ci-dessus ne seront pas utilisés. On trouvera dans la suite une description plus détaillée de ces composants, pouvant ou non être utilisés dans la présente Recommandation:

- La propriété d'adressage du message correspondant à l'adressage WS [action] DOIT contenir l'URI <http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify>.

Les composants du message *notify* sont décrits comme suit:

```
/wsnt:Notify
```

contient un ensemble comportant un ou plusieurs *NotificationMessages*:

/wsnt:Notify/wsnt:NotificationMessage

Ces derniers contiennent:

— la charge utile d'une notification:

/wsnt:Notify/wsnt:NotificationMessage/wsnt:SubscriptionReference

Il s'agit d'un *EndpointReference* de l'abonnement associé au message *notify*. Ce composant est facultatif.

/wsnt:Notify/wsnt:NotificationMessage/wsnt:ProducerReference

Il s'agit d'un *EndpointReference* du *NotificationProducer* ayant produit l'objet notification. Ce composant est facultatif.

/wsnt:Notify/wsnt:NotificationMessage/wsnt:Topic

Il s'agit d'un *TopicExpression* décrivant de manière exacte un thème, qui DOIT être le thème associé à la notification. Cet élément décrit le thème qui a été associé à un abonnement, à l'origine de l'envoi par le *NotificationProducer* du message *notify* au *NotificationConsumer*. De même, ce composant est facultatif.

/wsnt:Notify/wsnt:NotificationMessage/wsnt:Topic/@Dialect

Il s'agit du dialecte utilisé dans le *TopicExpression*. Il DOIT être le même que celui utilisé par l'abonné lors de la création de l'abonnement ayant conduit au message *notify* considéré.

/wsnt:Notify/wsnt:NotificationMessage/wsnt:Message

Il s'agit d'une copie de la charge utile effective de la notification. Ce composant peut être de n'importe quel type et sera remplacé par l'un des types de notification prédéfinis, définis au § A.5 et au sujet desquels des explications seront fournies plus avant.

et tout composant de métadonnées:

/wsnt:Notify/{any}

Les données brutes ne doivent pas être utilisées dans le cadre de la présente Recommandation.

Le format du message de notification le plus simplifié pouvant être utilisé dans le cadre de la présente Recommandation est indiqué dans le Tableau 5.

**Tableau 5 – Format de notification le plus simplifié utilisable dans le cadre de la présente Recommandation**

```
<wsnt:Notify>
  <wsnt:NotificationMessage>
    <wsnt:Message>
      {any}
    </wsnt:Message>
  </wsnt:NotificationMessage> +
</wsnt:Notify>
```

### 8.1.3.2 Définition de l'en-tête des notifications communes

On trouvera dans le présent paragraphe une description détaillée de l'en-tête des notifications communes, tel que décrit dans la Recommandation [UIT-T M.3702].

Le tableau suivant fournit la définition des paramètres de l'en-tête des notifications communes.

**Tableau 6 – Définition de l'en-tête des notifications communes**

Nom du paramètre	Qualificatifs	Type de données XSD	Descriptions
objectClass	M	xsd:string	Il s'agit du nom de la classe de l'objet géré qui envoie la notification. Voir le § 7.3.5 de [UIT-T M.3702]
objectInstance	M	x782:NameType	Il s'agit du DN de l'objet géré qui envoie la notification. Voir le § 7.3.5 de [UIT-T M.3702]
notificationId	M	782:NotificationIdType	Il s'agit de l'identificateur de la notification en vue d'établir des corrélations entre les notifications. Voir le § 7.3.5 de [UIT-T M.3702]
eventTime	M	xsd:dateTime	Il s'agit du moment auquel s'est produit l'événement. Voir le § 7.3.5 de [UIT-T M.3702]
systemDN	M	x782:NameType	Il s'agit du DN du système qui envoie la notification. Voir le § 7.3.5 de [UIT-T M.3702]
notificationType	M	nts:NotificationTypeType	<p>Ce paramètre indique le type de notification. Voir le § 7.3.5 de [UIT-T M.3702].</p> <p>Les types de notifications communes possibles définis dans la présente Recommandation peuvent être les suivants:</p> <ul style="list-style-type: none"> <li>– <i>attributeValueChange</i></li> <li>– <i>objectCreation</i></li> <li>– <i>objectDeletion</i></li> <li>– <i>stateChange</i></li> <li>– <i>communicationsAlarm</i></li> <li>– <i>environmentalAlarm</i></li> <li>– <i>equipmentAlarm</i></li> <li>– <i>processingErrorAlarm</i></li> <li>– <i>qualityOfServiceAlarm</i></li> <li>– <i>integrityViolation</i></li> <li>– <i>operationalViolation</i></li> <li>– <i>physicalViolation</i></li> <li>– <i>securityViolation</i></li> <li>– <i>timeDomainViolation</i></li> <li>– <i>relationshipChange</i></li> <li>– <i>heartbeat (pulsation)</i></li> </ul> <p>Cette liste peut être étendue afin d'inclure les nouveaux types de notifications lorsqu'ils sont définis.</p>

### 8.1.3.3 Définition du contenu des notifications communes

1) Contenu des notifications des types *objectCreation* et *objectDeletion*.

**Tableau 7 – Contenu des notifications des types *objectCreation* et *objectDeletion***

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.
sourceIndicator	O	x782:SourceIndicatorType	Cause de l'événement.
attributeList	O	x782:AttributeNameAndValueType	Valeurs des attributs.

2) Contenu des notifications du type *attributeValueChange*.

**Tableau 8 – Contenu des notifications du type *attributeValueChange***

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.
sourceIndicator	O	x782:SourceIndicatorType	Cause de l'événement.
attributeChanges	M	x782:AttributeValueChangeSetType	Attributs modifiés.

3) Contenu des notifications du type *stateChange*.

**Tableau 9 – Contenu des notifications du type *stateChange***

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.

**Tableau 9 – Contenu des notifications du type *stateChange***

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
sourceIndicator	O	x782:SourceIndicatorType	Cause de l'événement.
stateChanges	M	x782:AttributeValueChangeSetType	Etats modifiés.

4) Contenu des notifications des types *communicationAlarm*, *environmentalAlarm*, *equipmentAlarm*, *processingErrorAlarm* et *qualityOfServiceAlarm*.

**Tableau 10 – Contenu des notifications relatives à des alarmes**

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationSetType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.
probableCause	M	x782:ProbableCauseType	Cause probable de l'alarme.
specificProblems	O	x782:SpecificProblemSetType	Problèmes non normalisés.
perceivedSeverity	M	x782:PerceivedSeverityType	Ce paramètre indique la gravité perçue de l'alarme. Voir la définition des types pour plus d'informations.
backedUpStatus	O	xsd:boolean	"Vrai" si sauvegardée.
backUpObject	O	x782:NameType	Ce paramètre indique le DN de l'objet de sauvegarde si <i>backUpStatus</i> est "faux".
trendIndication	O	x782:TrendIndicationType	Voir la définition des types pour plus d'informations
thresholdInfo	O	x782:ThresholdInfoType	Voir la définition des types pour plus d'informations
stateChangeDefinition	O	x782:AttributeChangeSetType	Ce paramètre indique les changements d'état de l'alarme.
monitoredAttributes	O	x782:AttributeNameAndValueType	Voir la définition des types pour plus d'informations.
proposedRepairActions	O	x782:ProposedRepairActionSetType	Ce paramètre indique les actions proposées afin de corriger la défaillance.
alarmEffectOnService	O	xsd:boolean	"Vrai" si l'alarme a des incidences sur les services.

**Tableau 10 – Contenu des notifications relatives à des alarmes**

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
alarmingResumed	O	xsd:boolean	"Vrai" si l'émission de l'alarme vient d'être reprise, éventuellement à la suite d'un report de la signalisation.
suspectObjectList	O	x782:SuspectObjectSetType	Objets éventuellement impliqués dans la défaillance.

5) Contenu des notifications des types *integrityViolation*, *operationalViolation*, *physicalViolation*, *securityViolation* et *timeDomainViolation*.

**Tableau 11 – Contenu des notifications relatives à des transgressions**

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationSetType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.
securityAlarmCause	M	x782:SecurityAlarmCauseType	Cause de l'alarme de sécurité.
securityAlarmSeverity	M	x782:PerceivedSeverityType	Valeur <i>cleared</i> autorisée? La Recommandation UIT-T X.721 semble limiter l'utilisation de la valeur <i>cleared</i> pour cette alarme, mais cette valeur devrait être autorisée.
securityAlarmDetector	M	x782:SecurityAlarmDetectorType	Voir la définition des types pour plus d'informations.
serviceUser	M	x782:ServiceUserType	Utilisateur du service concerné par la transgression.
serviceProvider	M	x782:ServiceProviderType	Fournisseur du service concerné par la transgression.

6) Contenu des notifications du type *relationshipChange*.

**Tableau 12 – Contenu des notifications du type *relationshipChange***

Paramètre de la notification	Qualificatifs	Type de données	Descriptions
correlatedNotifications	O	x782:CorrelatedNotificationType	Liste des notifications corrélées.
additionalText	O	xsd:string	Message sous forme de texte.
additionalInfo	O	x782:AdditionalInformationSetType	Informations supplémentaires dans un format autre que du texte.
sourceIndicator	O	x782:SourceIndicatorType	Cause de l'événement.
relationshipChanges	M	x782:AttributeChangeSetType	Attributs de relation modifiés.

On trouvera les définitions détaillées des types de données concernant le contenu des notifications au § A.1.

## 8.2 Enregistrement de service UDDI

Le registre UDDI défini par l'OASIS est un service de répertoire fondé sur des services web.

L'UDDI (description, découverte et intégration universelles) repose sur la définition d'un ensemble de services assurant la description et la découverte des éléments suivants:

- entreprises, organisations et autres fournisseurs de services web;
- les services web qu'ils mettent à disposition; et
- les interfaces techniques qui peuvent être utilisées pour accéder à ces services.

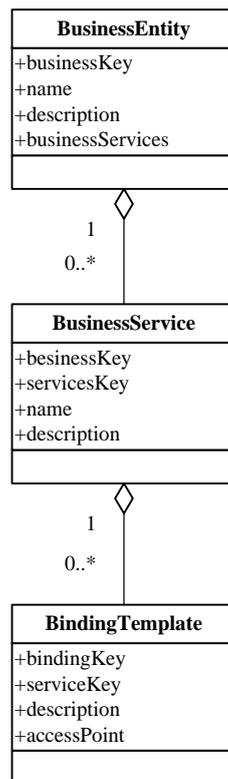
L'UDDI fournit une infrastructure interopérable constituant une base solide pour l'environnement d'un logiciel fondé sur des services web, autant pour des services accessibles au public que pour des services disponibles uniquement au sein d'une organisation.

Le registre UDDI permet à un client de trouver des services web parmi ceux qui sont répertoriés. Le cadre décrit utilise le service bien connu UDDI pour l'enregistrement de service et permet à un système gérant de découvrir les services fournis par un système géré.

Etant donné que l'UDDI n'est pas conçu spécialement pour la gestion de réseau, de nombreuses propriétés ne sont pas nécessaires lorsqu'il est utilisé pour la gestion de réseau.

Les paragraphes suivants fournissent un modèle d'informations UDDI simplifié et énumèrent certaines opérations connexes qui seront susceptibles d'être utilisées dans un domaine de gestion de réseau.

## 8.2.1 Modèle d'informations UDDI utilisé en gestion de réseau



### Modèle d'informations UDDI simplifié

Un modèle d'informations UDDI utilisé en gestion de réseau est constitué d'instances des types d'entités suivants:

- **businessEntity**: Décrit une entité du système fournissant des services de gestion. Il s'agit généralement d'un système géré (par exemple, un système EMS ou un agent) fourni par un vendeur. Dans certains cas, il peut aussi s'agir d'un système tiers fournissant des services autonomes destinés à être utilisés en partage par plusieurs systèmes gérés (par exemple, des services WSN). Dans un domaine de gestion de réseau, il s'agit généralement d'un système gérant utilisant les services fournis par différents systèmes gérés. Une instance donnée de la structure *businessEntity* est identifiée de façon univoque par son attribut *businessKey* dans le registre UDDI. Cet attribut est déterminé par le registre UDDI au moment de l'enregistrement. L'attribut *name* donne un surnom au système géré. Un système géré et un système tiers peuvent tous deux être enregistrés en tant qu'instances de *businessEntity*. Des informations textuelles simples relatives au système géré peuvent être indiquées dans l'attribut facultatif *description*. La liste des services fournis par ce système géré est contenue dans l'attribut *businessServices*. Les autres propriétés de *businessEntity* figurant dans le modèle UDDI d'origine ne sont pas utilisées dans le cadre décrit ici.
- **businessService**: Représente un service web logique assurant une fonctionnalité de gestion (par exemple, la gestion de la configuration, la gestion de la qualité de fonctionnement) au moyen du système géré. Il s'agit de l'unité permettant de regrouper les opérations de gestion étroitement liées entre elles (par exemple, obtenir ou modifier les valeurs des attributs des objets gérés). Une instance donnée de *businessService* est identifiée de façon univoque par son attribut *serviceKey* dans le registre UDDI. L'attribut *businessKey* identifie de manière univoque l'entité *businessEntity* qui contient et fournit le service en question. L'attribut *name* donne le surnom d'un service. Des informations textuelles simples relatives au service de gestion peuvent être indiquées dans l'attribut facultatif *description*, indiquant les

fonctionnalités et l'utilisation des services web, ainsi que d'autres informations descriptives à leur sujet. Les autres propriétés de *businessService* figurant dans le modèle UDDI d'origine ne sont pas utilisées dans le cadre décrit ici.

- **bindingTemplate:** Décrit les informations techniques dont a besoin le système gérant afin d'utiliser un service web particulier. Un *businessService* dans un domaine de gestion de réseau a généralement un seul *accessPoint*. Par conséquent, la relation entre un *businessService* et le *bindingTemplate* correspondant devrait être univoque, sauf dans les cas où la répartition de la charge ou la sauvegarde sont activées, lorsque plusieurs mises en oeuvre du service sont fournies. Les informations concernant chaque *accessPoint* sont alors contenues dans un *bindingTemplate*. Une entité de *bindingTemplate* donnée est identifiée de façon univoque par son attribut *bindingKey*. Il ne peut y avoir plus d'une mise en oeuvre assurant différentes fonctionnalités pour un service de gestion de réseau, car cela poserait problème lors de la tentative de sélection de l'*accessPoint* du service approprié par le système gérant. L'attribut *serviceKey* identifie de manière univoque le *businessService* qui contient le *bindingTemplate* considéré. Des informations textuelles simples relatives au *bindingTemplate* peuvent être indiquées dans l'attribut facultatif *description*. L'attribut *accessPoint* est une chaîne de caractères utilisée pour transmettre l'adresse de réseau permettant d'invoquer les services web.

## 8.2.2 Enregistrement d'un nouveau service et accès à ce service au moyen de l'UDDI

On trouve dans [OASIS UDDI] plusieurs interfaces API destinées autant aux fournisseurs de services web qu'aux clients de ces services et permettant d'enregistrer un service ou d'en trouver un. Les sous-paragraphe suivants décrivent certaines de ces interfaces API pouvant être utilisées dans un domaine de réseau.

### 8.2.2.1 Utilisation d'interfaces API de publication pour publier un nouveau service web dans le registre UDDI

Trois interfaces API d'opération peuvent être utilisées par un système géré (ou un tiers) pour enregistrer les services web fournis dans le registre UDDI. Elles sont énumérées dans le tableau suivant:

**Tableau 13 – Interfaces API d'opération pour l'enregistrement d'un service web dans le registre UDDI**

<b>Interfaces API d'opération</b>	<b>Direction du paramètre</b>	<b>Nom du paramètre</b>	<b>Descriptions</b>
get_authToken	arguments	userId	Cet attribut est un argument obligatoire et donne l'identificateur d'utilisateur attribué à un utilisateur autorisé individuel par un noeud UDDI. Les noeuds doivent permettre aux utilisateurs individuels d'obtenir un <i>userID</i> ainsi qu'une information de mot de passe qui seront valides au niveau du noeud considéré.
		cred	Cet attribut est un argument obligatoire comportant le mot de passe ou les informations d'identification associés à l'utilisateur.
	retour	authToken	En cas d'exécution réussie, cet appel API retourne une structure <i>authToken</i> contenant un élément <i>authInfo</i> valide pouvant être utilisé lors d'appels API ultérieurs nécessitant une valeur <i>authInfo</i> .
save_business:	arguments	authInfo	Cet argument est un élément contenant un jeton d'authentification. Les jetons d'authentification sont obtenus au moyen de l'appel API <i>get_authToken</i> . Les registres utilisés par plusieurs diffuseurs de publications et ceux qui appliquent des restrictions quant à l'auteur des publications requièrent généralement un argument <i>authInfo</i> pour cet appel.
		businessEntity	Élément récursif obligatoire contenant une ou plusieurs structures <i>businessEntity</i> .
	retour	businessKey	Valeur assignée suite à l'exécution de l'interface API <i>save_business</i> .
save_service	arguments	authInfo	Cet argument est un élément contenant un jeton d'authentification. Les jetons d'authentification sont obtenus au moyen de l'appel API <i>get_authToken</i> . Les registres utilisés par plusieurs diffuseurs de publications et ceux qui appliquent des restrictions quant à l'auteur des publications requièrent généralement un argument <i>authInfo</i> pour cet appel.
		businessService	Élément récursif obligatoire contenant un ou plusieurs éléments <i>businessService</i> complets.
	retour	serviceKey	Les valeurs de <i>serviceKey</i> et de <i>bindingKey</i> , assignées suite à l'exécution de l'interface API <i>save_service</i> , sont incluses dans les données de <i>businessService</i> .
		bindingKey	

### 8.2.2.2 Utilisation d'interfaces API de requête pour rechercher des services web dans le registre UDDI

**Tableau 14 – Interfaces API d'opération pour la recherche d'un service web dans le registre UDDI**

Interfaces API d'opération	Direction du paramètre	Nom du paramètre	Descriptions
find_business	arguments	name	Cet ensemble facultatif de chaînes de caractères représente un ou plusieurs <i>name</i> , éventuellement qualifiés par des attributs <i>xml:lang</i> . Etant donné que le comportement par défaut est <i>exactMatch</i> , la valeur fournie pour l'argument <i>name</i> doit être une correspondance exacte.
		findQualifiers	Cet ensemble d'éléments <i>findQualifier</i> peut être utilisé afin de modifier le comportement par défaut <i>exactMatch</i> de la fonctionnalité de recherche.
		maxRows	Cet entier facultatif permet au programme qui exécute la demande de fixer une limite pour le nombre de résultats retournés. Cet argument peut être utilisé conjointement avec l'argument <i>listHead</i> .
	retour	businessList	Retourne une <i>businessList</i> qui remplit les critères définis dans les arguments. <i>BusinessList</i> : cette structure est une séquence d'activités ( <i>business</i> ).
find_service	arguments	name	Cet ensemble facultatif de chaînes de caractères représente un ou plusieurs <i>name</i> , éventuellement qualifiés par des attributs <i>xml:lang</i> . Etant donné que le comportement par défaut est <i>exactMatch</i> , la valeur fournie pour l'argument <i>name</i> doit être une correspondance exacte.
		findQualifiers	Cet ensemble d'éléments <i>findQualifier</i> peut être utilisé afin de modifier le comportement par défaut <i>exactMatch</i> de la fonctionnalité de recherche.
		businessKey	Cette <i>uddi_key</i> est utilisée pour indiquer une instance particulière de <i>businessEntity</i> à rechercher. Cet argument est utilisé pour indiquer une <i>businessEntity</i> existante dans laquelle des services devraient être trouvés.
		maxRows	Cet entier facultatif permet au programme qui exécute la demande de fixer une limite pour le nombre de résultats retournés. Cet argument peut être utilisé conjointement avec l'argument <i>listHead</i> .
	retour	serviceList	Retourne une <i>serviceList</i> qui remplit les critères définis dans les arguments. <i>ServiceList</i> : cette structure est une séquence d'activités ( <i>business</i> ).

**Tableau 14 – Interfaces API d'opération pour la recherche d'un service web dans le registre UDDI**

Interfaces API d'opération	Direction du paramètre	Nom du paramètre	Descriptions
find_binding	arguments	serviceKey	Cette <i>uddi_key</i> facultative est utilisée pour indiquer une instance particulière d'un élément <i>businessService</i> dans les données enregistrées. La recherche est effectuée uniquement sur les relations ( <i>binding</i> ) des données du <i>businessService</i> particulier identifié par la <i>serviceKey</i> .
		findQualifier	Cet ensemble d'éléments <i>findQualifier</i> peut être utilisé afin de modifier le comportement par défaut <i>exactMatch</i> de la fonctionnalité de recherche.
		maxRows	Cet entier facultatif permet au programme qui exécute la demande de fixer une limite pour le nombre de résultats retournés. Cet argument peut être utilisé conjointement avec l'argument <i>listHead</i> .
	retour	bindingDetail	Retourne un <i>bindingDetail</i> contenant zéro, une ou plusieurs structures <i>bindingTemplate</i> qui remplissent les critères définis dans la liste des arguments. Chaque structure <i>bindingTemplate</i> comporte une <i>description</i> et un <i>accessPoint</i> .

## 9 Services supports du cadre

Le présent paragraphe définit les services supports communs qui sont inclus dans le cadre mais qui ne sont pas des services web bien connus. Ce paragraphe définit les services qui seront largement utilisés par les applications de gestion de réseau. Ces services offrent également les capacités requises pour permettre la réutilisation des modèles d'information existants sans modification notable de la sémantique. La description en langage WSDL des interfaces avec ces services est donnée dans l'Annexe A.

### 9.1 Service de pulsation

Le service de pulsation sert à vérifier le fonctionnement du mécanisme de transfert des notifications (c'est-à-dire des notifications WSN dans le cadre décrit) dans un système géré, ainsi que du réseau de communications entre le système géré et le système géant.

Il envoie périodiquement une petite notification à un système géant intéressé à la recevoir, qui identifie le système qui a émis la pulsation. Après la configuration de ce service, un système géant peut garantir le fonctionnement des notifications WSN. Comme ces notifications transitent par les mêmes logiciels et les mêmes réseaux que les notifications issues d'autres ressources, le fonctionnement de celles-ci est périodiquement vérifié.

Le service de pulsation comporte deux attributs internes, chacun d'eux étant constitué d'une paire de méthodes d'accès, une pour récupérer la valeur ("get") et l'autre pour la modifier ("set"), comme indiqué dans le tableau suivant:

**Tableau 15 – Attributs et opérations d'accès dans un service de pulsation**

Attributs internes des services de pulsation	Opérations	Message des demandes et des réponses
systemLabel	systemLabelGet	Request: NULL Response: name="systemLabel" type="hs:SystemLabelType"
	systemLabelSet	Request: name="systemLabel" type="hs:SystemLabelType" Response: NULL
period	periodGet	Request: NULL Response: name="period" type="hs:HeartbeatPeriodType"
	periodSet	Request: name="period" type="hs:HeartbeatPeriodType" Response: NULL

Le tableau suivant présente le contenu qui doit être inclus dans une notification de pulsation.

**Tableau 16 – Contenu d'une notification de pulsation**

Contenu d'une notification de pulsation	Type de données	Descriptions
systemLabel	xsd:string	Identifie le système géré ayant envoyé la notification de pulsation.
period	xsd:unsigned Long	Indique l'intervalle de temps séparant deux pulsations.
timeStamp	xsd:dateTime	Marqueur temporel correspondant au moment auquel la notification de pulsation a été générée.

Les attributs *systemLabel* et *period* sont deux attributs d'un service de pulsation. L'attribut *systemLabel* est un identificateur fourni par l'utilisateur, dont l'usage prévu est de permettre à un système gérant d'insérer une étiquette désignant le système qui fournit la pulsation. La valeur de *systemLabel* peut être modifiée à l'aide de l'opération *systemLabelSet*. L'attribut *period* correspond à l'intervalle de temps séparant deux pulsations. La valeur de cet attribut peut être déterminée au moyen de l'opération *periodSet*. La valeur indiquée pour cette opération est l'intervalle de temps, en secondes, pendant laquelle le service de pulsation attend entre chaque émission de notification.

Chaque notification comporte la valeur de l'attribut *systemLabel*, la valeur actuelle de *period* et l'attribut *timeStamp*.

**(R) PULSATION-1:** Si le service de pulsation est assuré par un système géré, celui-ci doit prendre en charge l'interface de pulsation décrite ci-dessus et définie en WSDL au § A.2. La capacité décrite ci-dessus doit être prise en charge.

**(R) PULSATION-2:** La mise à jour de *period* doit faire que le service achemine une notification vers le service WSN contenant la nouvelle valeur de l'intervalle *period* puis commence à appliquer ce nouvel intervalle. La mise à zéro de l'intervalle fait que le service émet une seule notification finale comportant une valeur d'intervalle égale à zéro, puis l'arrêt de ces émissions (jusqu'à la réinitialisation de *period*).

**(R) PULSATION-3:** Les notifications de pulsation doivent être envoyées au service WSN une seule fois par intervalle de temps jusqu'à un changement de *period*. L'intervalle d'envoi des notifications de pulsation à un service WSN ne doit jamais être supérieur à deux fois la valeur de *period*.

## 9.2 Service d'opérations sur objets multiples

L'interface avec le service d'opérations sur objets multiples (MOO) est définie au § A.3 et est, en termes de programmation, faiblement typée. Elle offre un ensemble de capacités génériques qui peuvent être invoquées au sujet d'ensembles de types quelconques d'objets gérés (de type quelconque). Les opérations prises en charge sont les suivantes:

- Obtention limitée à un domaine: renvoie les valeurs de chacun des objets pour une liste d'attributs.
- Mise à jour limitée à un domaine: remplace une valeur d'attribut ou ajoute/supprime des valeurs dans des attributs à valeur d'ensemble. Cette opération peut servir à mettre à jour un ou plusieurs attributs dans un seul objet ou dans plusieurs objets.
- Suppression limitée à un domaine: supprime plusieurs objets.

Un service de base doit seulement prendre en charge l'opération d'obtention limitée à un domaine. Les deux autres opérations sont facultatives.

### 9.2.1 Paramètres communs aux opérations du service d'opérations MOO

Chacune des opérations limitées à un domaine exige quatre paramètres pour définir l'ensemble des objets sur lesquels l'opération sera effectuée.

- Nom d'objet de base: nom de l'objet situé à la racine d'un arbre d'objets sur lequel l'opération pourra être effectuée.
- Domaine: un *complexType* définissant les objets contenus par l'objet de base sur lequel l'opération pourra être effectuée. Ce *complexType* comporte deux éléments, l'un d'eux, *scope*, indiquant le domaine parmi quatre cas de figure possibles, dont deux contiennent un entier spécifiant un niveau d'objets contenus au-dessous de l'objet de base, représenté par le second élément *level* du *complexType*. Les quatre cas sont les suivants:
  - objet de base seulement. Si *scope* a la valeur *baseObjectOnly*, seul l'objet cible (de base) nommé est inclus dans ce domaine. Dans ce cas, l'élément *level* n'est pas utilisé;
  - sous-arborescence entière. Si *scope* a la valeur *wholeSubtree*, le domaine englobe tous les objets contenus au-dessous de l'objet de base, y compris celui-ci;
  - niveau individuel. Si *scope* a la valeur *individualLevel*, l'entier positif *level* sera aussi utilisé. Tous les objets contenus à un niveau inférieur à l'objet de base égal à cette valeur sont dans le domaine. Les objets directement contenus par l'objet de base sont au niveau 1;
  - base à niveau. Si *scope* a la valeur *baseToLevel*, l'entier positif *level* sera aussi utilisé. Le domaine inclura tous les objets en descendant de la base jusqu'au niveau indiqué, y compris l'objet de base et l'objet du niveau indiqué.

La signature XSD du *scopeType* est la suivante:

**Tableau 17 – Définition du *ScopeType***

ScopeEnumType	<pre>&lt;xsd:simpleType name="ScopeEnumType"&gt;   &lt;xsd:restriction base="xsd:string"&gt;     &lt;xsd:enumeration value="BasicObjectOnly"/&gt;     &lt;xsd:enumeration value="WholeSubtree"/&gt;     &lt;xsd:enumeration value="IndividualLevel"/&gt;     &lt;xsd:enumeration value="BaseToLevel"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre>
ScopeType	<pre>&lt;xsd:complexType name="ScopeType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="scopeInd" type="moos:ScopeEnumType"/&gt;     &lt;xsd:element name="level" type="xsd:short"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>

Les noms d'objets sont au format *NameType* (DN), tel que décrit dans la Recommandation [UIT-T X.782]. Le domaine est un *complexType* dont les valeurs sont indiquées ci-dessus.

Les paragraphes qui suivent donnent de plus amples détails sur chacune des opérations limitées à un domaine.

### 9.2.2 Obtention limitée à un domaine

La signature WSDL de l'opération d'obtention limitée à un domaine est la suivante dans le service d'opérations MOO de base:

**Tableau 18 – Demande et réponse pour l'opération *ScopedGet***

ScopedGet Request	<pre>&lt;xsd:complexType name="ScopedGetRequestType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="baseName" type="x782:NameType"/&gt;     &lt;xsd:element name="scope" type="moos:ScopeType"/&gt;     &lt;xsd:element name="moClassList" type="x782:MOClassListType" minOccurs="0" maxOccurs="1"/&gt;     &lt;xsd:element name="attributes" type="x782:StringSetType"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
ScopedGet Response	<pre>&lt;xsd:complexType name="ScopedGetResponseType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="moInfo" type="moos:GetResultsType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>

Comme décrit ci-dessus, les deux premiers paramètres *baseName* et *scope* dans la demande *scopedGet* servent à sélectionner un ensemble d'objets sur lesquels l'opération d'obtention sera effectuée. Le paramètre facultatif *moClassList* dans la demande *scopedGet* peut servir à sélectionner les objets d'une ou plusieurs classes MOC particulières. Pour chacun des objets, le service d'opérations MOO essaiera de renvoyer une valeur relative à chacun des attributs nommés dans le paramètre *attributes*, qui n'est qu'une liste de chaînes. Une liste d'attributs vides a, lorsqu'elle est soumise, la signification spéciale que les valeurs de tous les attributs doivent être renvoyées pour les objets sélectionnés. Les types de données contenus dans la valeur de retour sont donnés dans le tableau suivant:

**Tableau 19 – Types de données pour *ScopedGet***

X782:AttributeNameAndValueType	<pre>&lt;xsd:complexType name="AttributeNameAndValueType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="attributeName" type="xsd:string"/&gt;     &lt;xsd:element name="attributeType" type="xsd:string"/&gt;     &lt;xsd:element name="attributeValue" type="x782:AttributeValueType"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
X782:AttributeNameAndValueSetType	<pre>&lt;xsd:complexType name="AttributeNameAndValueSetType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="attributeNameAndValue" type="x782:AttributeNameAndValueType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
GetResultsType	<pre>&lt;xsd:complexType name="GetResultsType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="name" type="x782:NameType"/&gt;     &lt;xsd:element name="attributes" type="x782:AttributeNameAndValueSetType"/&gt;     &lt;xsd:element name="failedAttributes" type="x782:StringSetType"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
X782:StringSetType	<pre>&lt;xsd:complexType name="StringSetType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="value" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>

Les deux premiers types `x782:AttributeNameAndValueType` et `x782:AttributeNameAndValueSetType` forment une liste de paires nom-valeur (*name-value*). Le type de retour est un *complexType* contenant une séquence de structures individuelles pour chaque objet géré sélectionné. Chaque structure contient un nom d'objet géré, la liste des valeurs d'attribut issues de cet objet et les noms des éventuels attributs qui n'ont pas pu être extraits de cet objet. Si une valeur d'attribut d'un objet géré ne peut pas être extraite, soit parce que l'objet n'a pas d'attribut correspondant ou parce qu'une exception a été levée au sujet de l'accès, le nom de cet attribut doit être placé sur la liste des attributs ayant conduit à un échec pour cet objet.

### 9.2.3 Mise à jour limitée à un domaine

La signature WSDL de l'opération de mise à jour limitée à un domaine est la suivante dans le service d'opérations MOO:

**Tableau 20 – Demande et réponse pour l'opération *ScopedUpdate***

ScopedUpdate Request	<pre> &lt;wsdl:message name="scopedUpdateRequest"&gt;   &lt;wsdl:part name="scopedUpdateInput" type="moos:ScopedUpdateRequestType"/&gt; &lt;/wsdl:message&gt; &lt;xsd:complexType name="ScopedUpdateRequestType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="baseName" type="x782:NameType"/&gt;     &lt;xsd:element name="scope" type="moos:ScopeType"/&gt;     &lt;xsd:element name="moClassList" type="x782:MOClassListType" minOccurs="0" maxOccurs="1"/&gt;     &lt;xsd:element name="modifications" type="moas:AttributeNVMLListType"/&gt;     &lt;xsd:element name="failuresOnly" type="xsd:boolean"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; </pre>
ScopedUpdate Response	<pre> &lt;wsdl:message name="scopedUpdateResponse"&gt;   &lt;wsdl:part name="ScopedUpdateOutput" type="moos:ScopedUpdateResponseType"/&gt; &lt;/wsdl:message&gt; &lt;xsd:complexType name="ScopedUpdateResponseType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="updateResult" type="moos:UpdateResultsType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; </pre>

Ici encore, les deux premiers paramètres servent à sélectionner l'ensemble des objets dont la mise à jour doit être effectuée. Le paramètre facultatif *moClassList* peut servir à sélectionner les objets d'une ou plusieurs classes MOC particulières. Les modifications prennent la forme d'une séquence de structures accompagnées chacune du nom d'un attribut, de son type, d'une valeur pour cet attribut et d'une valeur énumérée indiquant si la valeur doit remplacer la valeur actuelle de l'attribut, doit être ajoutée à la valeur actuelle de l'attribut ou doit en être retranchée. Les options *ADDValues* et *REMOVEValues* ne sont valides que si le type de l'attribut est un type complexe XSD d'une séquence et que les valeurs de l'attribut peuvent être ajoutées ou soustraites. Les valeurs contenues dans la séquence de modifications des structures sont acheminées sous forme de *anyTypes* XML. Les signatures XSD pour les types de données relatifs aux modifications sont indiquées dans le tableau suivant:

**Tableau 21 – Types de données relatifs aux modifications pour l'opération *ScopedUpdate***

moas:ModifyOptionType	<pre>&lt;xsd:simpleType name="ModifyOptionType"&gt;   &lt;xsd:restriction base="xsd:string"&gt;     &lt;xsd:enumeration value="REPLACE"/&gt;     &lt;xsd:enumeration value="ADDValues"/&gt;     &lt;xsd:enumeration value="REMOVEValues"/&gt;     &lt;xsd:enumeration value="SETToefault"/&gt;   &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre>
moas:AttributeNVMTType	<pre>&lt;xsd:complexType name="AttributeNVMTType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="attributeName" type="xsd:string"/&gt;     &lt;xsd:element name="attributeType" type="xsd:string"/&gt;     &lt;xsd:element name="attributeValue" type="x782:AttributeValueType"/&gt;     &lt;xsd:element name="modifyOption" type="moas:ModificationOpType"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
moas:AttributeNVMListType	<pre>&lt;xsd:complexType name="AttributeNVMListType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="attributeNVM" type="moas: AttributeNVMTType" minOccurs="1" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>

Le fanion *failuresOnly* sert à indiquer si le client souhaite que le service retourne des résultats pour tous les objets relevant du domaine ou seulement pour les objets pour lesquels au moins une des modifications n'a pas pu être exécutée bien qu'ils relèvent du domaine.

La valeur de retour est une séquence de structures contenant chacune un nom d'objet ainsi qu'une liste des éventuels attributs qui n'ont pas pu être modifiés. Le service essaiera d'exécuter dans l'ordre toutes les modifications indiquées dans la liste et continuera cet essai même si une modification échoue au sujet d'un attribut, auquel cas le nom de celui-ci est ajouté à la liste des échecs. Si l'élément de données *failedAttributes* est vide, le client sait que toutes les mises à jour ont été effectuées sur cet objet. Les nouveaux types intervenant dans la valeur de retour sont indiqués dans le tableau suivant:

**Tableau 22 – Type de données du retour pour l'opération *ScopedUpdate***

UpdateResultsType	<pre>&lt;xsd:complexType name="UpdateResultsType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="name" type="x782:NameType" minOccurs="1" maxOccurs="unbounded"/&gt;     &lt;xsd:element name="failedAttributes" type="x782:StringSetType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
-------------------	--

## 9.2.4 Suppression limitée à un domaine

La signature WSDL de l'opération de suppression limitée à un domaine est la suivante dans le service d'opérations MOO évolué:

**Tableau 23 – Demande et réponse pour l'opération *ScopedDelete***

ScopedDelete Request	<pre>&lt;wsdl:message name="scopedDeleteRequest"&gt;   &lt;wsdl:part name="scopedDeleteInput" type="nts:ScopedDeleteRequestType"/&gt; &lt;/wsdl:message&gt; &lt;xsd:complexType name="ScopedDeleteRequestType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="baseName" type="x782:NameType"/&gt;     &lt;xsd:element name="scope" type="moos:ScopeType"/&gt;     &lt;xsd:element name="moClassList" type="x782:MOClassListType" minOccurs="0" maxOccurs="1"/&gt;     &lt;xsd:element name="failuresOnly" type="xsd:boolean"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
ScopedDelete Response	<pre>&lt;wsdl:message name="scopedDeleteResponse"&gt;   &lt;wsdl:part name="scopedDeleteOutput" type="moos:ScopedDeleteResponseType"/&gt; &lt;/wsdl:message&gt; &lt;xsd:complexType name="ScopedDeleteResponseType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="deleteResult" type="moos: DeleteResultsType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>

Cette opération tente simplement de supprimer chaque objet sélectionné selon le paramètre *scope*. Le paramètre facultatif *moClassList* peut servir à sélectionner les objets d'une ou plusieurs classes MOC particulières.

Le fanion *failuresOnly* sert à indiquer si le client souhaite que le service retourne des résultats pour tous les objets relevant du domaine ou seulement pour les objets qui n'ont pas pu être supprimés. Etant donné que des notifications de suppression d'objet sont normalement envoyées, les clients peuvent souvent décider de ne recevoir de résultats que pour les objets qui n'ont pas pu être supprimés.

La valeur de retour énumère les noms de tous les objets avec un fanion, *notDeletable*, qui doit être "vrai" si l'objet sélectionné n'a pas pu être supprimé, soit en raison de sa politique de suppression soit parce qu'il a levé une exception.

Les types de données intervenant dans la valeur de retour sont indiqués dans le tableau suivant:

**Tableau 24 – Types de données pour l'opération *ScopedDelete***

DeleteResults Type	<pre>&lt;xsd:complexType name="DeleteResultsType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="name" type="x782:NameType" minOccurs="1" maxOccurs="unbounded"/&gt;     &lt;xsd:element name="notDeletable" type="xsd:boolean" minOccurs="0" maxOccurs="1"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt;</pre>
--------------------	---

Etant donné qu'un grand nombre d'objets ne peuvent pas être supprimés s'ils contiennent d'autres objets, le service doit commencer à supprimer les objets feuilles relevant du domaine et à progresser vers l'objet racine dans les domaines fondés sur des relations de contenance. Lors de la suppression d'objets, le service d'opérations MOO doit toujours suivre les règles de suppression d'objet fondées sur la politique de suppression d'objet. Comme ces règles vont être appliquées à chacun des objets relevant du domaine en allant de bas en haut, l'effet sera toutefois différent d'un simple essai de suppression d'objet à la racine d'une sous-arborescence. De même, le service d'opérations MOO est de type "au mieux". Il est donc possible que certains des objets d'un sous-arbre relevant du domaine soient supprimés alors que d'autres ne le sont pas. Les règles qui doivent toujours être appliquées aux opérations de suppression limitées à un domaine sont les suivantes:

- 1) aucun objet ne peut être laissé à l'abandon. Autrement dit, un objet ne peut pas être supprimé sans suppression de tous les objets (subordonnés) qu'il contient;
- 2) l'exécution d'une opération de suppression limitée à un domaine sur un sous-arbre entier a comme résultat la suppression de tous les objets contenus dans ce sous-arbre, à moins qu'un objet ne soit pas supprimable ou qu'un objet ait un subordonné qui n'est pas supprimable;
- 3) l'exécution d'une opération de suppression limitée à un domaine sur une partie d'un sous-arbre nécessite l'évaluation de chacun des objets situés au plus bas niveau du domaine. Si un objet situé au plus bas niveau du domaine peut être supprimé, cet objet est supprimé ainsi que ses éventuels subordonnés. Si un objet situé au plus bas niveau ne peut pas être supprimé, il ne l'est pas, non plus que l'un quelconque de ses objets supérieurs. D'autres objets relevant du domaine peuvent cependant être supprimés si les règles de suppression le permettent. Le service remonte ensuite à la couche suivante, et ainsi de suite.

### 9.2.5 Exigences du service d'opérations MOO

Ce paragraphe résume les exigences relatives au service d'opérations sur objets multiples.

**(R) MOO-1:** Une mise en oeuvre du service d'opérations MOO doit prendre en charge l'opération *scopedGet* décrite ci-dessus et dont la définition en WSDL figure au § A.3.

**(O) MOO-2:** A titre d'option, une mise en oeuvre du service d'opérations MOO peut prendre en charge les opérations *scopedUpdate* et *ScopedDelete* décrites ci-dessus et dont les définitions en WSDL figurent au § A.3.

## 9.3 Service de contenance

Dans le domaine de la gestion de réseau, une fonction doit pouvoir signaler les objets qui sont contenus dans un objet supérieur, afin de vérifier qu'un tel objet existe avant qu'un subordonné soit créé, afin de s'assurer que l'on ne crée pas deux objets portant le même nom, etc. Le cadre sera élargi afin de prendre en charge cette fonction par l'ajout d'un nouveau service: le service de contenance.

### 9.3.1 Description du service de contenance

La principale fonction qui doit être prise en charge par le service de contenance est de permettre à un système gérant d'interroger un système géré avec le nom d'un objet et de recevoir en retour le nom des objets contenus par cet objet. L'on définira également un moyen d'obtenir les noms ajoutés au service et retranchés de celui-ci. Ces noms ne sont pas à l'usage des systèmes gérants mais à l'usage interne des objets gérés, des ateliers et d'autres parties d'un système géré. Ils sont fournis afin de faciliter la mise au point de composants réutilisables, éventuellement par des tierces parties, et sont définis à une interface distincte de celle qui est utilisée par les systèmes gérants.

### 9.3.2 Définition du service de contenance

Le service de contenance offre trois opérations permettant d'extraire les informations de contenance. La description en WSDL de l'interface du service de contenance et le schéma XML correspondant sont décrits au § A.4.

Une brève description de la sémantique du service de contenance est fournie dans le tableau suivant:

**Tableau 25 – Opérations du service de contenance**

Opération	Direction du paramètre	Nom du paramètre	Descriptions
exists	demande	name: x782:NameType	Il s'agit du DN d'un objet géré dont il faut vérifier l'existence dans le service de contenance.
	réponse	existsOutput : xsd:boolean	Renvoie "vrai" si l'objet géré indiqué existe dans le service de contenance; renvoie "faux" sinon.
getContained	demande	base: x782:NameType	Indique l'instance de l'objet géré de base pour un arbre particulier dont on souhaite obtenir les informations de contenance.
		scope: ScopeType	Indique les informations relatives au domaine pour la récupération des informations de contenance. Voir le § 9.2.1 pour la sémantique relative au domaine.
	réponse	moList: x782:NameSet Type	Renvoie une liste contenant le nom des objets gérés spécifiés par les paramètres <i>base</i> et <i>scope</i> .
getContainedBy Class	demande	base : x782:NameType	Indique l'instance de l'objet géré de base pour un arbre particulier dont on souhaite obtenir les informations de contenance.
		moClass: xsd:string	Indique la classe d'objets gérés dont on souhaite obtenir les informations de contenance.
		scope: moos:ScopeType	Indique les informations relatives au domaine pour la récupération des informations de contenance. Voir le § 9.2.1 pour la sémantique relative au domaine.
	réponse	moList: x782:NameSet Type	Renvoie une liste contenant le nom des objets gérés spécifiés par les paramètres <i>base</i> et <i>scope</i> et qui sont des instances de la classe indiquée par le paramètre <i>moClass</i> .

L'opération *exists* prend un nom (*name*) et renvoie la valeur "vrai" si ce nom est inscrit dans le service de contenance. Les deux autres opérations renvoient le nom des objets contenus dans l'objet désigné dans le paramètre *base*. Le paramètre *scope* peut être utilisé dans ces deux opérations afin de spécifier la partie à extraire de l'arbre d'objets contenu au-dessous de l'objet de base. La troisième opération, *getContainedByClass*, prend un paramètre *moClass* afin de demander au service de contenance de renvoyer le nom des objets appartenant à une classe MOC déterminée.

**(R) CONTENANCE-1:** L'interface prise en charge par le service de contenance doit être l'interface de contenance décrite ci-dessus et définie au § A.4.

**(R) CONTENANCE-2:** En réponse à une invocation de l'opération *exists*, le service de contenance doit renvoyer la valeur "vrai" si le nom est déjà inscrit au service et la valeur "faux" dans le cas contraire. Si une erreur de serveur empêche cette détermination, une exception d'erreur d'application appropriée doit être renvoyée au système gérant.

**(R) CONTENANCE-3:** En réponse à une invocation de l'opération *getContained*, le service de contenance doit renvoyer la liste des noms des objets contenus dans l'objet nommé dans le paramètre *base*. La liste des objets contenus doit être déterminée conformément au paramètre *scope*. Si le nom de l'élément de base donné est vide, le premier niveau des noms contenus doit correspondre aux noms des objets gérés enregistrés de la racine. Si une erreur empêche le renvoi de la liste, une exception d'erreur d'application appropriée doit être transmise, par exemple, si le nom de base n'est pas enregistré.

**(R) CONTENANCE-4:** Le service de contenance doit répondre à l'invocation de l'opération *getContainedByClass* comme décrit dans la prescription CONTENANCE-3, sauf que seuls les noms qui correspondent au paramètre *moClass* doivent être renvoyés.

L'enregistrement des noms des objets gérés dans le service de contenance est une fonction qui doit être assurée par les systèmes gérés et qui n'entre pas dans le cadre de la présente Recommandation.

## 10 Observance et conformité

Le présent paragraphe définit les critères qui doivent être satisfaits par les autres documents de normalisation revendiquant l'observance du cadre décrit ici. Il définit également les fonctions qui doivent être mises en oeuvre par les systèmes revendiquant la conformité à la présente Recommandation.

### 10.1 Points de conformité du système

Le présent paragraphe résume les fonctions individuelles qui ont été décrites plus haut dans la présente Recommandation. Ces points de conformité seront ensuite combinés en profils qui devront être pris en charge par les systèmes revendiquant la conformité à la présente Recommandation.

- 1) Une mise en oeuvre revendiquant la conformité aux exigences relatives aux notifications doit:
  - prendre en charge la version de la notification de base des services web [OASIS WSN] indiquée au § 7.2;
  - prendre en charge l'interface de notification décrite au § 8.1 et définie en WSDL et sous forme de schéma XML au § A.1;
  - prendre en charge le format de notification décrit au § 8.1.3 et défini sous forme de schéma XML au § A.5.
- 2) Une mise en oeuvre revendiquant la conformité aux exigences relatives au registre de services doit:
  - prendre en charge la version de l'enregistrement de service UDDI [OASIS UDDI] indiquée au § 7.2;
  - prendre en charge l'enregistrement de service UDDI tel qu'indiqué au § 8.2.
- 3) Une mise en oeuvre revendiquant la conformité au service de pulsation doit:
  - prendre en charge l'interface du service de pulsation décrite au § 9.1 et définie en WSDL et sous forme de schéma XML au § A.2;
  - répondre aux exigences relatives au service de pulsation indiquées au § 9.1.
- 4) Une mise en oeuvre revendiquant la conformité au service d'opérations sur objets multiples (MOO) doit:
  - prendre en charge l'interface du service d'opérations MOO décrite au § 9.2 et définie en WSDL et sous forme de schéma XML au § A.3;
  - répondre à toutes les exigences obligatoires relatives au service d'opérations MOO indiquées au § 9.2.5.

- 5) Une mise en oeuvre revendiquant la conformité au service de contenance doit:
- prendre en charge l'interface du service de contenance décrite au § 9.3 et définie en WSDL et sous forme de schéma XML au § A.4;
  - répondre aux exigences relatives au service de contenance indiquées au § 9.2.

## **10.2 Profil de conformité de base**

Un système revendiquant la conformité au profil de base UIT-T Q.818 doit prendre en charge:

- 1) les versions du langage WSDL, des schémas XML et du protocole SOAP indiquées au § 7.2;
- 2) la notification WSN (voir le point de conformité 1);
- 3) l'enregistrement de service UDDI (voir le point de conformité 2);
- 4) le service de pulsation (voir le point de conformité 3);
- 5) le service d'opérations MOO (voir le point de conformité 5);
- 6) le service de contenance (voir le point de conformité 4).

## Annexe A

### Définition en WSDL des services supports du cadre

(Cette annexe fait partie intégrante de la présente Recommandation.)

#### A.1 Définition en WSDL et sous forme de schéma XML du service de notification de l'UIT

##### 1) Définition sous forme de schéma XML du service de notification de l'UIT

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Schema Definition for data types to be used in Notification Service
specified in this Recommendation.
  Filename : q818_NotificationService.xsd -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
  xmlns:nts="http://www.itu.int/xml-namespace/itu-t/q.818/NotificationService"
  targetNamespace="http://www.itu.int/xml-namespace/itu-
t/q.818/NotificationService"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="1.0">

<xsd:import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
  schemaLocation="x782.xsd"/>

  <xsd:complexType name="FilterType">
    <xsd:sequence>
      <xsd:element name="language" type="xsd:string"/>
      <xsd:any minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="EndpointReferenceType" mixed="false">
    <xsd:sequence>
      <xsd:element name="address" type="nts:AttributedURIType"/>
      <xsd:element name="referenceParameters"
type="nts:ReferenceParametersType" minOccurs="0"/>
      <xsd:element name="metadata" type="nts:MetadataType" minOccurs="0"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="AttributedURIType" mixed="false">
    <xsd:simpleContent>
      <xsd:extension base="xsd:anyURI">
        <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:complexType name="ReferenceParametersType" mixed="false">
    <xsd:sequence>
```

```

        <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="MetadataType" mixed="false">
    <xsd:sequence>
        <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>

<xsd:simpleType name="StatusType">
    <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>
<xsd:simpleType name="IdType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:complexType name="IdSetType">
    <xsd:sequence>
        <xsd:element name="id" type="nts:IdType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="SubscriptionStatusType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="locked"/>
        <xsd:enumeration value="unlocked"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="SubscribeRequestType">
    <xsd:sequence>
        <xsd:element name="managerId" type="nts:IdType"/>
        <xsd:element name="notificationTypes"
type="nts:NotificationTypeListType"/>
        <xsd:element name="filteringCriteria" type="nts:FilterType"
minOccurs="0"/>
        <xsd:element name="destination" type="nts:EndpointReferenceType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SubscribeResponseType">
    <xsd:sequence>
        <xsd:element name="subscriptionId" type="nts:IdType"/>
        <xsd:element name="status" type="nts:StatusType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UnsubscribeRequestType">
    <xsd:sequence>
        <xsd:element name="managerId" type="nts:IdType"/>
        <xsd:element name="subscriptionId" type="nts:IdType"/>
    </xsd:sequence>

```

```

</xsd:complexType>
<xsd:complexType name="UnsubscribeResponseType">
  <xsd:sequence>
    <xsd:element name="status" type="nts:StatusType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SuspendSubscriptionRequestType">
  <xsd:sequence>
    <xsd:element name="managerId" type="nts:IdType"/>
    <xsd:element name="subscriptionId" type="nts:IdType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SuspendSubscriptionResponseType">
  <xsd:sequence>
    <xsd:element name="status" type="nts:StatusType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ResumeSubscriptionRequestType">
  <xsd:sequence>
    <xsd:element name="managerId" type="nts:IdType"/>
    <xsd:element name="subscriptionId" type="nts:IdType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ResumeSubscriptionResponseType">
  <xsd:sequence>
    <xsd:element name="status" type="nts:StatusType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetNotificationTypesRequestType">
  <xsd:sequence>
    <xsd:element name="notificationIRPIId" type="x782:NameType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetNotificationTypesResponseType">
  <xsd:sequence>
    <xsd:element name="notificationTypeList"
type="nts:NotificationTypeListType"/>
    <xsd:element name="status" type="nts:StatusType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="QuerySubscriptionRequestType">
  <xsd:sequence>
    <xsd:element name="subscriptionId" type="nts:IdType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="QuerySubscriptionResponseType">
  <xsd:sequence>
    <xsd:element name="notificationTypeList"
type="nts:NotificationTypeListType"/>
    <xsd:element name="subscriptionStatus"
type="nts:SubscriptionStatusType"/>
  </xsd:sequence>

```

```

        <xsd:element name="filteringCriteria" type="nts:FilterType"/>
        <xsd:element name="destination" type="nts:EndpointReferenceType"/>
        <xsd:element name="status" type="nts:StatusType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ModifySubscriptionRequestType">
    <xsd:sequence>
        <xsd:element name="subscriptionId" type="xsd:string"/>
        <xsd:element name="filteringCriteria" type="nts:FilterType"
minOccurs="0"/>
        <xsd:element name="destination" type="nts:EndpointReferenceType"
minOccurs="0"/>
        <xsd:element name="notificationTypes" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ModifySubscriptionResponseType">
    <xsd:sequence>
        <xsd:element name="status" type="nts:StatusType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ListAllSubscriptionIdsRequestType">
    <xsd:sequence>
        <xsd:element name="managerId" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ListAllSubscriptionIdsResponseType">
    <xsd:sequence>
        <xsd:element name="subscriptionIdSet" type="nts:IdSetType"/>
        <xsd:element name="status" type="nts:StatusType"/>
    </xsd:sequence>
</xsd:complexType>

```

<!-- The following provides the XML Schema definitions for the common notifications contents defined in clause 8.3.1.2 and 8.3.1.3 of

this Recommendation. The common data types referenced in this Annex is from [ITU-T X.782]. -->

```

<xsd:simpleType name="NotificationTypeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="objectCreation"/>
        <xsd:enumeration value="objectDeletion"/>
        <xsd:enumeration value="attributeValueChange"/>
        <xsd:enumeration value="stateChange"/>
        <xsd:enumeration value="communicationsAlarm"/>
        <xsd:enumeration value="environmentalAlarm"/>
        <xsd:enumeration value="equipmentAlarm"/>
        <xsd:enumeration value="processingErrorAlarm"/>
        <xsd:enumeration value="qualityOfServiceAlarm"/>
        <xsd:enumeration value="integrityViolation"/>
        <xsd:enumeration value="operationalViolation"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

        <xsd:enumeration value="physicalViolation"/>
        <xsd:enumeration value="securityViolation"/>
        <xsd:enumeration value="timeDomainViolation"/>
        <xsd:enumeration value="relationshipChange"/>
        <xsd:enumeration value="heartbeat"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="NotificationTypeListType">
    <xsd:sequence>
        <xsd:element name="notificationType" type="nts:NotificationTypeType"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<!-- The following is the common notification header content definition -->
<xsd:complexType name="CommonNotificationHeaderType">
    <xsd:sequence>
        <xsd:element name="objectClass" type="xsd:string"/>
        <xsd:element name="objectInstance" type="x782:NameType"/>
        <xsd:element name="notificationID" type="x782:NotificationIDType"/>
        <xsd:element name="eventTime" type="xsd:dateTime"/>
        <xsd:element name="systemDN" type="x782:NameType"/>
        <xsd:element name="notificationType" type="nts:NotificationTypeType"/>
    </xsd:sequence>
</xsd:complexType>

<!-- The following is the notification content definition for objectCreation
and objectDeletion-->
<xsd:complexType name="ObjectCreationDeletionNotificationType">
    <xsd:sequence>
        <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
        <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="sourceIndicator" type="x782:SourceIndicatorType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="attributeList" type="x782:AttributeNameAndValueType"
minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<!-- The following is the notification content definition for stateChange -->
<xsd:complexType name="StateChangeNotificationType">
    <xsd:sequence>
        <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
        <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationSetType" minOccurs="0" maxOccurs="1"/>

```

```

        <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="sourceIndicator" type="x782:SourceIndicatorType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="stateChanges" type="x782:AttributeChangeSetType"/>
    </xsd:sequence>
</xsd:complexType>

```

<!-- The following is the notification content definition for attributeValueChange -->

```

<xsd:complexType name="AttributeValueChangeNotificationType">
    <xsd:sequence>
        <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
        <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="sourceIndicator" type="x782:SourceIndicatorType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="attributeChanges"
type="x782:AttributeChangeSetType"/>
    </xsd:sequence>
</xsd:complexType>

```

<!-- The following is the notification content definition for "communicationAlarm", "environmentalAlarm", "equipmentAlarm",

"processingErrorAlarm", "qualityOfServiceAlarm" -->

```

<xsd:complexType name="AlarmNotificationType">
    <xsd:sequence>
        <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
        <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="probableCause" type="x782:ProbableCauseType"/>
        <xsd:element name="specificProblems" type="x782:SpecificProblemSetType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="perceivedSeverity"
type="x782:PerceivedSeverityType"/>
        <xsd:element name="backedUpStatus" type="xsd:boolean" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="backedUpObject" type="x782:NameType" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="trendIndication" type="x782:TrendIndicationType"
minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="thresholdInfo" type="x782:ThresholdInfoType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="stateChangeDefinition"
type="x782:AttributeChangeSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="monitoredAttributes"
type="x782:AttributeNameAndValueType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="proposedRepairActions"
type="x782:ProposedRepairActionSetType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="alarmEffectOnService" type="xsd:boolean"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="alarmingResumed" type="xsd:boolean" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="suspectObjectList" type="x782:SuspectObjectSetType"
minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

```

<!-- The following is the notification content definition for "integrityViolation", "operationalViolation", "physicalViolation",

"securityViolation", "timeDomainViolation" -->

```

    <xsd:complexType name="ViolationNotificationType">
        <xsd:sequence>
            <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
            <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationSetType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="securityAlarmCause"
type="x782:SecurityAlarmCauseType"/>
            <xsd:element name="securityAlarmSeverity"
type="x782:PerceivedSeverityType"/>
            <xsd:element name="securityAlarmDetector"
type="x782:SecurityAlarmDetectorType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="serviceUser" type="x782:ServiceUserType"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="serviceProvider" type="x782:ServiceProviderType"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="thresholdInfo" type="x782:ThresholdInfoType"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="stateChangeDefinition"
type="x782:AttributeChangeSetType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="monitoredAttributes"
type="x782:AttributeNameAndValueType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="proposedRepairActions"
type="x782:ProposedRepairActionSetType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="alarmEffectOnService" type="xsd:boolean"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="alarmingResumed" type="xsd:boolean" minOccurs="0"
maxOccurs="1"/>
            <xsd:element name="suspectObjectList" type="x782:SuspectObjectSetType"
minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </complexType>

```

```

</xsd:complexType>

<!-- The following is the notification content definition for
"relationshipChange" -->
<xsd:complexType name="RelationshipChangeNotificationType">
  <xsd:sequence>
    <xsd:element name="notificationHeader"
type="nts:CommonNotificationHeaderType"/>
    <xsd:element name="correlatedNotifications"
type="x782:CorrelatedNotificationSetType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="additionalText" type="x782:AdditionalTextType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="additionalInfo"
type="x782:AdditionalInformationSetType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="sourceIndicator" type="x782:SourceIndicatorType"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="RelationshipChanges"
type="x782:AttributeChangeSetType"/>
  </xsd:sequence>
</xsd:complexType>

<!-- The following is the notification content definition for "heartbeat" -->
<xsd:complexType name="HeartbeatNotificationType">
  <xsd:sequence>
    <xsd:element name="systemLabel" type="xsd:string"/>
    <xsd:element name="period" type="xsd:unsignedLong"/>
    <xsd:element name="timeStamp" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

## 2) Définition en WSDL du service de notification de l'UIT

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL Operation Definition for Notification Service specified in this
Recommendation.
Filename : q818_NotificationService.wsdl -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:nts="http://www.itu.int/xml-namespace/itu-t/q.818/NotificationService"
name="NotificationService" targetNamespace="http://www.itu.int/xml-
namespace/itu-t/q.818/NotificationService">
  <import namespace="http://www.itu.int/xml-namespace/itu-
t/q.818/NotificationService" location="q818_NotificationService.xsd"/>
  <wsdl:message name="subscribeRequest">
    <wsdl:part name="subscribeInput" type="nts:SubscribeRequestType"/>
  </wsdl:message>
  <wsdl:message name="subscribeResponse">
    <wsdl:part name="subscribeOutput" type="nts:SubscribeResponseType"/>
  </wsdl:message>
  <wsdl:message name="unsubscribeRequest">
    <wsdl:part name="unsubscribeInput" type="nts:UnsubscribeRequestType"/>
  </wsdl:message>

```

```

</wsdl:message>
<wsdl:message name="unsubscribeResponse">
  <wsdl:part name="unsubscribeOutput" type="nts:UnsubscribeResponseType"/>
</wsdl:message>
<wsdl:message name="suspendSubscriptionRequest">
  <wsdl:part name="suspendSubscriptionInput"
type="nts:SuspendSubscriptionRequestType"/>
</wsdl:message>
<wsdl:message name="suspendSubscriptionResponse">
  <wsdl:part name="suspendSubscriptionOutput"
type="nts:SuspendSubscriptionResponseType"/>
</wsdl:message>
<wsdl:message name="resumeSubscriptionRequest">
  <wsdl:part name="resumeSubscriptionInput"
type="nts:ResumeSubscriptionRequestType"/>
</wsdl:message>
<wsdl:message name="resumeSubscriptionResponse">
  <wsdl:part name="resumeSubscriptionOutput"
type="nts:ResumeSubscriptionResponseType"/>
</wsdl:message>
<wsdl:message name="getNotificationTypesRequest">
  <wsdl:part name="getNotificationTypesInput"
type="nts:GetNotificationTypesRequestType"/>
</wsdl:message>
<wsdl:message name="getNotificationTypesResponse">
  <wsdl:part name="getNotificationTypesOutput"
type="nts:GetNotificationTypesResponseType"/>
</wsdl:message>
<wsdl:message name="querySubscriptionRequest">
  <wsdl:part name="querySubscriptionInput"
type="nts:QuerySubscriptionRequestType"/>
</wsdl:message>
<wsdl:message name="querySubscriptionResponse">
  <wsdl:part name="querySubscriptionOutput"
type="nts:QuerySubscriptionResponseType"/>
</wsdl:message>
<wsdl:message name="modifySubscriptionRequest">
  <wsdl:part name="modifySubscriptionInput"
type="nts:ModifySubscriptionRequestType"/>
</wsdl:message>
<wsdl:message name="modifySubscriptionResponse">
  <wsdl:part name="modifySubscriptionOutput"
type="nts:ModifySubscriptionResponseType"/>
</wsdl:message>
<wsdl:message name="listAllSubscriptionIdsRequest">
  <wsdl:part name="listAllSubscriptionIdsInput"
type="nts:ListAllSubscriptionIdsRequestType"/>
</wsdl:message>
<wsdl:message name="listAllSubscriptionIdsResponse">
  <wsdl:part name="listAllSubscriptionIdsOutput"
type="nts:ListAllSubscriptionIdsResponseType"/>
</wsdl:message>
<wsdl:portType name="NotificationService">

```

```

<wsdl:operation name="subscribe">
  <wsdl:input message="nts:subscribeRequest"/>
  <wsdl:output message="nts:subscribeResponse"/>
</wsdl:operation>
<wsdl:operation name="unsubscribe">
  <wsdl:input message="nts:unsubscribeRequest"/>
  <wsdl:output message="nts:unsubscribeResponse"/>
</wsdl:operation>
<wsdl:operation name="suspendSubscription">
  <wsdl:input message="nts:suspendSubscriptionRequest"/>
  <wsdl:output message="nts:suspendSubscriptionResponse"/>
</wsdl:operation>
<wsdl:operation name="resumeSubscription">
  <wsdl:input message="nts:resumeSubscriptionRequest"/>
  <wsdl:output message="nts:resumeSubscriptionResponse"/>
</wsdl:operation>
<wsdl:operation name="getNotificationTypes">
  <wsdl:input message="nts:getNotificationTypesRequest"/>
  <wsdl:output message="nts:getNotificationTypesResponse"/>
</wsdl:operation>
<wsdl:operation name="querySubscription">
  <wsdl:input message="nts:querySubscriptionRequest"/>
  <wsdl:output message="nts:querySubscriptionResponse"/>
</wsdl:operation>
<wsdl:operation name="modifySubscription">
  <wsdl:input message="nts:modifySubscriptionRequest"/>
  <wsdl:output message="nts:modifySubscriptionResponse"/>
</wsdl:operation>
<wsdl:operation name="listAllSubscriptionIds">
  <wsdl:input message="nts:listAllSubscriptionIdsRequest"/>
  <wsdl:output message="nts:listAllSubscriptionIdsResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="NotificationServiceBinding"
type="nts:NotificationService">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="subscribe">
    <soap:operation soapAction="subscribe"/>
    <wsdl:input>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="unsubscribe">
    <soap:operation soapAction=" unsubscribe "/>
    <wsdl:input>

```

```

        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="suspendSubscription">
            <soap:operation soapAction="suspendSubscription" />
            <wsdl:input>
                <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                </wsdl:input>
            <wsdl:output>
                <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="resumeSubscription">
                <soap:operation soapAction="resumeSubscription" />
                <wsdl:input>
                    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                    </wsdl:input>
                <wsdl:output>
                    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                    </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getNotificationTypes">
                    <soap:operation soapAction="getNotificationTypes" />
                    <wsdl:input>
                        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                        </wsdl:input>
                    <wsdl:output>
                        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                        </wsdl:output>
                    </wsdl:operation>
                    <wsdl:operation name="querySubscription">
                        <soap:operation soapAction="querySubscription" />
                        <wsdl:input>
                            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                            </wsdl:input>
                        <wsdl:output>
                            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
                            </wsdl:output>
                        </wsdl:operation>
                        <wsdl:operation name="modifySubscription">

```

```

        <soap:operation soapAction="modifySubscription"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="listAllSubscriptionIds">
        <soap:operation soapAction="listAllSubscriptionIds"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="NotificationService">
    <wsdl:port name="NotificationService"
binding="nts:NotificationServiceBinding">
        <soap:address location="http://www.itu.int/xml-namespace/itu-
t/q.818/NotificationService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## A.2 Définition en WSDL et sous forme de schéma XML du service de pulsation de l'UIT

### 1) Définition sous forme de schéma XML du service de pulsation de l'UIT

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Schema Definition for data types to be used in HeartbeatService
specified in this Recommendation.

Filename : q818_HeartbeatService.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:hs="http://www.itu.int/xml-namespace/itu-t/q.818/HeartbeatService"
targetNamespace="http://www.itu.int/xml-namespace/itu-t/q.818/HeartbeatService"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xsd:simpleType name="SystemLabelType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="HeartbeatPeriodType">
        <xsd:restriction base="xsd:unsignedLong"/>
    </xsd:simpleType>
    <xsd:simpleType name="GeneralizedTimeType">

```

```

        <xsd:restriction base="xsd:dateTime"/>
    </xsd:simpleType>
</xsd:schema>

```

## 2) Définition en WSDL du service de pulsation de l'UIT

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL Operation Definition for Heartbeat Service specified in this
Recommendation.

    Filename : q818_HeartbeatService.wsdl -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:hs="http://www.itu.int/xml-namespace/itu-t/q.818/HeartbeatService"
name="HeartbeatService" targetNamespace="http://www.itu.int/xml-namespace/itu-
t/q.818/HeartbeatService">
    <import namespace="http://www.itu.int/xml-namespace/itu-
t/q.818/HeartbeatService" location="q818_HeartbeatService.xsd"/>
    <wsdl:message name="periodMessage">
        <wsdl:part name="period" type="hs:HeartbeatPeriodType"/>
    </wsdl:message>
    <wsdl:message name="systemLabelMessage">
        <wsdl:part name="systemLabel" type="hs:SystemLabelType"/>
    </wsdl:message>
    <wsdl:portType name="HeartbeatServicePort">
        <wsdl:operation name="periodGet">
            <wsdl:output message="hs:periodMessage"/>
        </wsdl:operation>
        <wsdl:operation name="periodSet">
            <wsdl:input message="hs:periodMessage"/>
        </wsdl:operation>
        <wsdl:operation name="systemLabelGet">
            <wsdl:output message="hs:systemLabelMessage"/>
        </wsdl:operation>
        <wsdl:operation name="systemLabelSet">
            <wsdl:input message="hs:systemLabelMessage"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="HeartbeatServiceBinding" type="hs:HeartbeatServicePort">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="periodGet">
            <soap:operation soapAction="periodGet"/>
            <wsdl:output>
                <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>

```

```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="periodSet">
        <soap:operation soapAction="periodSet"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="systemLabelGet">
        <soap:operation soapAction="systemLabelGet"/>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="systemLabelSet">
        <soap:operation soapAction="systemLabelSet"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </wsdl:input>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="HeartbeatService">
    <wsdl:port name="HeartbeatServicePort"
binding="hs:HeartbeatServiceBinding">
        <soap:address location="http://www.itu.int/xml-namespace/itu-
t/q.818/HeartbeatService"/>
    </wsdl:port>
</wsdl:service>
<!-- The contents of heartbeat notification is defined in Annex A.1, and it
is to be sent by OASIS WSN Service, not by HeartbeatService itself. -->
</wsdl:definitions>

```

### **A.3 Définition en WSDL et sous forme de schéma XML du service d'opérations MOO de l'UIT**

#### **1) Définition sous forme de schéma XML du service d'opérations MOO de l'UIT**

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Schema Definition for data types to be used in Multiple Object
Operation(MOO) Service specified in this Recommendation.
Filename : q818_MOOService.xsd -->

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
xmlns:moas="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"
xmlns:moos="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService" targetNamespace="http://www.itu.int/xml-
namespace/itu-t/q.818/MultipleObjectOperationService"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
  <xsd:import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
schemaLocation="x782.xsd"/>
  <xsd:import namespace="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService" schemaLocation="x782_MOAccessService.xsd"/>
  <xsd:simpleType name="ScopeEnumType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="BasicObjectOnly"/>
      <xsd:enumeration value="WholeSubtree"/>
      <xsd:enumeration value="IndividualLevel"/>
      <xsd:enumeration value="BaseToLevel"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="ScopeType">
    <xsd:sequence>
      <xsd:element name="scopeInd" type="moos:ScopeEnumType"/>
      <xsd:element name="level" type="xsd:short" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="GetResultsType">
    <xsd:sequence>
      <xsd:element name="name" type="x782:NameType"/>
      <xsd:element name="attributes"
type="x782:AttributeNameAndValueSetType"/>
      <xsd:element name="failedAttributes" type="x782:StringSetType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="UpdateResultsType">
    <xsd:sequence>
      <xsd:element name="name" type="x782:NameType" minOccurs="1"
maxOccurs="unbounded"/>
      <xsd:element name="failedAttributes" type="x782:StringSetType"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="DeleteResultsType">
    <xsd:sequence>
      <xsd:element name="name" type="x782:NameType" minOccurs="1"
maxOccurs="unbounded"/>

```

```

        <xsd:element name="notDeletable" type="xsd:boolean" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ScopedGetRequestType">
    <xsd:sequence>
        <xsd:element name="baseName" type="x782:NameType"/>
        <xsd:element name="scope" type="moos:ScopeType"/>
        <xsd:element name="moClassList" type="x782:MOClassListType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="attributes" type="x782:StringSetType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ScopedGetResponseType">
    <xsd:sequence>
        <xsd:element name="moInfo" type="moos:GetResultsType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ScopedUpdateRequestType">
    <xsd:sequence>
        <xsd:element name="baseName" type="x782:NameType"/>
        <xsd:element name="scope" type="moos:ScopeType"/>
        <xsd:element name="moClassList" type="x782:MOClassListType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="modifications" type="moas:AttributeNVMLListType"/>
        <xsd:element name="failuresOnly" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ScopedUpdateResponseType">
    <xsd:sequence>
        <xsd:element name="updateResult" type="moos:UpdateResultsType"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ScopedDeleteRequestType">
    <xsd:sequence>
        <xsd:element name="baseName" type="x782:NameType"/>
        <xsd:element name="scope" type="moos:ScopeType"/>
        <xsd:element name="moClassList" type="x782:MOClassListType"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="failuresOnly" type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="ScopedDeleteResponseType">
  <xsd:sequence>
    <xsd:element name="deleteResult" type="moos:DeleteResultsType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## 2) Définition en WSDL du service d'opérations MOO de l'UIT

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL Operation Definition for Multiple Object Operation(MOO) Service
specified in this Recommendation.
  Filename : q818_MOOService.wsdl -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:moos="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService" name="MOOService"
targetNamespace="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService">
  <import namespace="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService" location="q818_MOOService.xsd"/>
  <wsdl:message name="scopedGetRequest">
    <wsdl:part name="scopedGetInput" type="moos:ScopedGetRequestType"/>
  </wsdl:message>
  <wsdl:message name="scopedGetResponse">
    <wsdl:part name="scopedGetOutput" type="moos:ScopedGetResponseType"/>
  </wsdl:message>
  <wsdl:message name="scopedUpdateRequest">
    <wsdl:part name="scopedUpdateInput" type="moos:ScopedUpdateRequestType"/>
  </wsdl:message>
  <wsdl:message name="scopedUpdateResponse">
    <wsdl:part name="scopedUpdateOutput"
type="moos:ScopedUpdateResponseType"/>
  </wsdl:message>
  <wsdl:message name="scopedDeleteRequest">
    <wsdl:part name="scopedDeleteInput" type="moos:ScopedDeleteRequestType"/>
  </wsdl:message>
  <wsdl:message name="scopedDeleteResponse">
    <wsdl:part name="scopedDeleteOutput"
type="moos:ScopedDeleteResponseType"/>
  </wsdl:message>
  <wsdl:portType name="MOOServicePort">
    <wsdl:operation name="scopedGet">

```

```

        <wsdl:input message="moos:scopedGetRequest"/>
        <wsdl:output message="moos:scopedGetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="scopedUpdate">
        <wsdl:input message="moos:scopedUpdateRequest"/>
        <wsdl:output message="moos:scopedUpdateResponse"/>
    </wsdl:operation>
    <wsdl:operation name="scopedDelete">
        <wsdl:input message="moos:scopedDeleteRequest"/>
        <wsdl:output message="moos:scopedDeleteResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="MOOServiceBinding" type="moos:MOOServicePort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="scopedGet">
        <soap:operation soapAction="scopedGet"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="scopedUpdate">
        <soap:operation soapAction="scopedUpdate"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="scopedDelete">
        <soap:operation soapAction="scopedDelete"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
        </wsdl:input>
        <wsdl:output>

```

```

        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="MOOService">
    <wsdl:port name="MOOServicePort" binding="moos:MOOServiceBinding">
        <soap:address location="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## **A.4 Définition en WSDL et sous forme de schéma XML du service de contenance de l'UIT**

### **1) Définition sous forme de schéma XML du service de contenance de l'UIT**

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Schema Definition for Containment Service to be used in this
Recommendation.
    Filename : q818_ContainmentService.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:cs="http://www.itu.int/xml-namespace/itu-t/q.818/ContainmentService"
    xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
    xmlns:moos="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService"
    targetNamespace="http://www.itu.int/xml-namespace/itu-
t/q.818/ContainmentService"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0">
    <xsd:import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
schemaLocation="x782.xsd"/>
    <xsd:import namespace="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService" schemaLocation="q818_MOOService.xsd"/>
    <xsd:complexType name="GetContainedRequestType">
        <xsd:sequence>
            <xsd:element name="base" type="x782:NameType"/>
            <xsd:element name="scope" type="moos:ScopeType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="GetContainedByClassRequestType">
        <xsd:sequence>
            <xsd:element name="base" type="x782:NameType"/>
            <xsd:element name="class" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

```

```

        <xsd:element name="scope" type="moos:ScopeType"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## 2) Définition en WSDL du service de contenance de l'UIT

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL Operation Definition for Containment Service specified in this
Recommendation.

    Filename : q818_ContainmentService.wsdl -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
xmlns:cs="http://www.itu.int/xml-namespace/itu-t/q.818/ContainmentService"
xmlns:ns="http://www.itu.int/xml-namespace/itu-
t/q.818/MultipleObjectOperationService" xmlns:ns1="http://www.itu.int/xml-
namespace/itu-t/x.782/MOAccessService" name="ContainmentService"
targetNamespace="http://www.itu.int/xml-namespace/itu-
t/q.818/ContainmentService">
    <import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
location="x782.xsd"/>
    <import namespace="http://www.itu.int/xml-namespace/itu-
t/q.818/ContainmentService" location="q818_ContainmentService.xsd"/>

    <wsdl:message name="existsRequest">
        <wsdl:part name="name" type="x782:NameType"/>
    </wsdl:message>
    <wsdl:message name="existsResponse">
        <wsdl:part name="existsOutput" type="xsd:boolean"/>
    </wsdl:message>
    <wsdl:message name="getContainedRequest">
        <wsdl:part name="getContainedInput" type="cs:GetContainedRequestType"/>
    </wsdl:message>
    <wsdl:message name="getContainedResponse">
        <wsdl:part name="moList" type="x782:NameSetType"/>
    </wsdl:message>
    <wsdl:message name="getContainedByClassRequest">
        <wsdl:part name="getContainedByClassInput"
type="cs:GetContainedByClassRequestType"/>
    </wsdl:message>
    <wsdl:message name="getContainedByClassResponse">
        <wsdl:part name="moList" type="x782:NameSetType"/>
    </wsdl:message>
    <wsdl:portType name="Containment">

```

```

<wsdl:operation name="exists">
  <wsdl:input message="cs:existsRequest"/>
  <wsdl:output message="cs:existsResponse"/>
</wsdl:operation>
<wsdl:operation name="getContained">
  <wsdl:input message="cs:getContainedRequest"/>
  <wsdl:output message="cs:getContainedResponse"/>
</wsdl:operation>
<wsdl:operation name="getContainedByClass">
  <wsdl:input message="cs:getContainedByClassRequest"/>
  <wsdl:output message="cs:getContainedByClassResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ContainmentServiceSoapBinding" type="cs:Containment">
  <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="exists">
    <soap:operation soapAction="exists"/>
    <wsdl:input name="existRequest">
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:input>
    <wsdl:output name="existResponse">
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getContained">
    <soap:operation soapAction="getContained"/>
    <wsdl:input name="getContainedRequest">
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:input>
    <wsdl:output name="getContainedResponse">
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getContainedByClass">
    <soap:operation soapAction="getContainedByClass"/>
    <wsdl:input name="getContainedByClassRequest">

```

```
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:input>
    <wsdl:output name="getContainedByClassResponse">
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:ContainmentService-pcs"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ContainmentService">
    <wsdl:port name="ContainmentService"
binding="cs:ContainmentServiceSoapBinding">
        <soap:address location="http://www.itu.int/xml-namespace/itu-
t/q.818/ContainmentService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

## Bibliographie

- [b-UIT-T Q.816] Recommandation UIT-T Q.816 (2001), *Services RGT à architecture CORBA*.
- [b-UIT-T Q.816.1] Recommandation UIT-T Q.816.1 (2001), *Services RGT à architecture CORBA: extensions pour la prise en charge des interfaces à granularité grossière*.
- [b-UIT-T Q.816.2] Recommandation UIT-T Q.816.2 (2007), *Services RGT à architecture CORBA: extensions pour la prise en charge des interfaces orientées service*.





## SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série D	Principes de tarification et de comptabilité et questions de politique générale et d'économie relatives aux télécommunications internationales/TIC
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Environnement et TIC, changement climatique, déchets d'équipements électriques et électroniques, efficacité énergétique; construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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
<b>Série Q</b>	<b>Commutation et signalisation et mesures et tests associés</b>
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, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet, réseaux de prochaine génération, Internet des objets et villes intelligentes
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication