



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.781

(08/2001)

SÉRIE X: RÉSEAUX DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS
Gestion OSI – Fonctions de gestion et fonctions ODMA

**Spécifications et directives pour l'établissement
de formulaires de déclaration de conformité
d'implémentations associés aux systèmes de
type CORBA**

Recommandation UIT-T X.781

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS DE DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

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

Recommandation UIT-T X.781

Spécifications et directives pour l'établissement de formulaires de déclaration de conformité d'implémentations associés aux systèmes de type CORBA

Résumé

La présente Recommandation spécifie le formulaire de déclaration de conformité d'implémentation (ICS) des interfaces système de type CORBA qui seront utilisées pour la gestion des réseaux de télécommunication; elle définit également la méthode de test à utiliser pour les interfaces de type CORBA.

Source

La Recommandation X.781 de l'UIT-T, élaborée par la Commission d'études 4 (2001-2004) de l'UIT-T, a été approuvée le 13 août 2001 selon la procédure définie dans la Résolution 1 de l'AMNT.

AVANT-PROPOS

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

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.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2002

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	Page	
1	Domaine d'application	1
2	Références normatives	1
3	Définitions	1
3.1	Définition de déclaration de conformité d'implémentation	1
4	Abréviations et acronymes.....	2
5	Fondement de la méthode de test de conformité pour les interfaces de système de gestion CORBA	2
5.1	Généralités	2
5.2	Méthodologie de test de conformité pour les interfaces de type CORBA.....	2
6	Formulaire ICS pour interface de gestion de type CORBA	3
6.1	Description générale de l'IDL	4
6.2	Prescriptions et directives de spécification des formulaires CIICS	4
6.2.1	Instructions générales de la spécification des formulaires CIICS	5
6.2.2	Formulaire de prise en charge d'interface.....	5
6.2.3	Formulaire de prise en charge d'attribut	6
6.2.4	Formulaire de prise en charge d'opérations	7
6.2.5	Formulaire de prise en charge de type de données.....	8
7	Instructions pour remplir le formulaire CIICS	9
7.1	Définition du terme "Support"	9
7.2	Colonne "Status"	9
7.3	Colonne "prise en charge"	10
7.4	Colonne "Constraints and Values"	10
7.5	Colonne "Additional Information"	11
7.6	Colonne "Index"	12
7.7	Colonne "Subindex"	12
7.8	Colonne "Category"	12
7.9	Colonne "Propriété du champ"	12
7.10	Colonnes "Interface identifier", "Attribute identifier" et "Operation identifier"	12
7.11	Colonne "Derived interface"	13
	Annexe A – Exemple de spécification de formulaire CIICS.....	13
A.1	Définition IDL CORBA.....	13
A.2	CIICS	13

Annexe B – Exemple de spécification de "formulaire de prise en charge de type de données"	17
B.1 Définition IDL CORBA.....	17
B.2 CIICS	17
Annexe C – Exemple de spécification de "formulaire de prise en charge d'appendice de type quelconque"	19
C.1 Définition CORBA IDL.....	19
C.2 CIICS	20
Annexe D – Exemple de spécification du "formulaire de prise en charge de valuetype"	21
D.1 Définition CORBA IDL.....	21
D.2 Déclaration CIICS.....	22

Recommandation UIT-T X.781

Spécifications et directives pour l'établissement de formulaires de déclaration de conformité d'implémentations associés aux systèmes de type CORBA

1 Domaine d'application

La présente Recommandation définit les spécifications et les directives pour l'établissement de formulaires de déclaration de conformité d'implémentations d'interface CORBA (CIICS, *CORBA-based interface implementation conformance statement*) et spécifie le formulaire. La déclaration CIICS permet à un réalisateur de déclarer la conformité à une définition d'interface de type CORBA/IDL.

2 Références normatives

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

- UIT-T X.296 (1995), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Déclarations de conformité d'instance.*
- UIT-T X.724 (1996), *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'implémentations associés à la gestion OSI.*
- ISO/CEI 9646-7:1995, *Technologies de l'information – Interconnexion des systèmes ouverts (OSI) – Essais de conformité – Méthodologie générale et procédures – Partie 7: Déclarations de conformité des mises en œuvre.*
- OMG (1998), *The Common Object Request Broker: Architecture and Specification, Revision 2.3.*

3 Définitions

3.1 Définition de déclaration de conformité d'implémentation

La présente Recommandation utilise les termes ci-après, définis dans la Rec. UIT-T X.296 et l'ISO/CEI 9646-7:

- a) élément (ICS);
- b) question (ICS);
- c) (valeur de) descripteur d'état;
- d) réponse (support).

4 Abréviations et acronymes

La présente Recommandation utilise les abréviations suivantes:

CIICS	déclaration de conformité d'implémentation d'interface CORBA (<i>CORBA-based interface implementation conformance statement</i>)
CORBA	architecture de courtier commun de requêtes sur des objets (<i>common object request broker architecture</i>)
ICS	déclaration de conformité d'implémentation (<i>implementation conformance statement</i>)
IDL	langage de définition d'interface (<i>interface definition language</i>)
IUT	implémentation sous test (<i>implementation under test</i>)
IXIT	information supplémentaire d'implémentation pour le test (<i>implementation extra information for testing</i>)
ODP	traitement réparti ouvert (<i>open distributed processing</i>)

5 Fondement de la méthode de test de conformité pour les interfaces de système de gestion CORBA

5.1 Généralités

La conformité associe une implémentation à une norme. Elle indique dans quelle mesure il peut y avoir des variantes de systèmes, mis en œuvre conformément à une même norme, fonctionnant correctement en association. Lorsqu'une implémentation répond à ces prescriptions, elle est conforme à la norme. La vérification des déclarations constitue le test de conformité. Le point de départ est la définition de prescriptions de conformité dans des spécifications d'interface indépendantes de l'implémentation sur la base d'une identification de points de référence. Une spécification d'interface de gestion doit définir les points de référence de conformité au niveau desquels un objet doit être testé pour vérifier s'il respecte un ensemble de critères de conformité. Pendant le test, un certain nombre de stimuli et d'événements sont observés et évalués au niveau de ces points de conformité.

Des spécifications d'interface de gestion doivent inclure les déclarations de conformité qui identifient les points de référence de conformité au niveau de chaque interface d'objets spécifiés.

Etant donné qu'en général, le flux d'informations entre deux composantes de système est assuré via plusieurs points de référence, le test de conformité doit prendre en considération:

- a) le test de ces flux d'informations en chaque point de référence,
- b) le test de cohérence entre les combinaisons.

Ainsi, un test coordonné au niveau de tous les points de référence identifiés est nécessaire.

5.2 Méthodologie de test de conformité pour les interfaces de type CORBA

Il existe des prescriptions de tests de conformité statique et de conformité dynamique.

- a) Les prescriptions de test de conformité statique indiquent la fonctionnalité dont la présence est au minimum nécessaire pour le test de conformité. La base d'un test de conformité statique est un formulaire ICS dans lequel les limites et possibilités fonctionnelles de la norme sont définies. Il s'agit d'un document sous forme d'un questionnaire auquel doit répondre le réalisateur.
- b) Les prescriptions de test de conformité dynamique spécifient l'ensemble des comportements potentiels d'une réalisation visible aux points de référence identifiés. Le test de conformité dynamique intègre l'exécution des tests élémentaires dans un système de test.

Les tests élémentaires incluent les tests élémentaires concrets et les tests élémentaires abstraits. Les tests élémentaires concrets sont issus des tests élémentaires abstraits qui doivent être indiqués ou définis par la norme. La condition préalable de base pour la détermination des tests élémentaires est la définition des objectifs de test par la norme.

Avant d'entreprendre les tests dans un laboratoire, il faut procéder par étapes comme suit:

- a) le réalisateur doit remplir le formulaire ICS donné. L'information qui rend les déclarations de formulaire ICS plus précises ou concerne la matérialisation du test est prise en compte dans les informations IXIT (informations supplémentaires sur l'application destinées au test);
- b) les tests élémentaires réels (concrets) doivent découler des tests élémentaires abstraits de la norme, ou découler de l'objet des tests directement, où des entités génériques (types, etc.) sont réalisées en fonction de la réalisation individuelle.

La Figure 5-1 illustre cette approche.

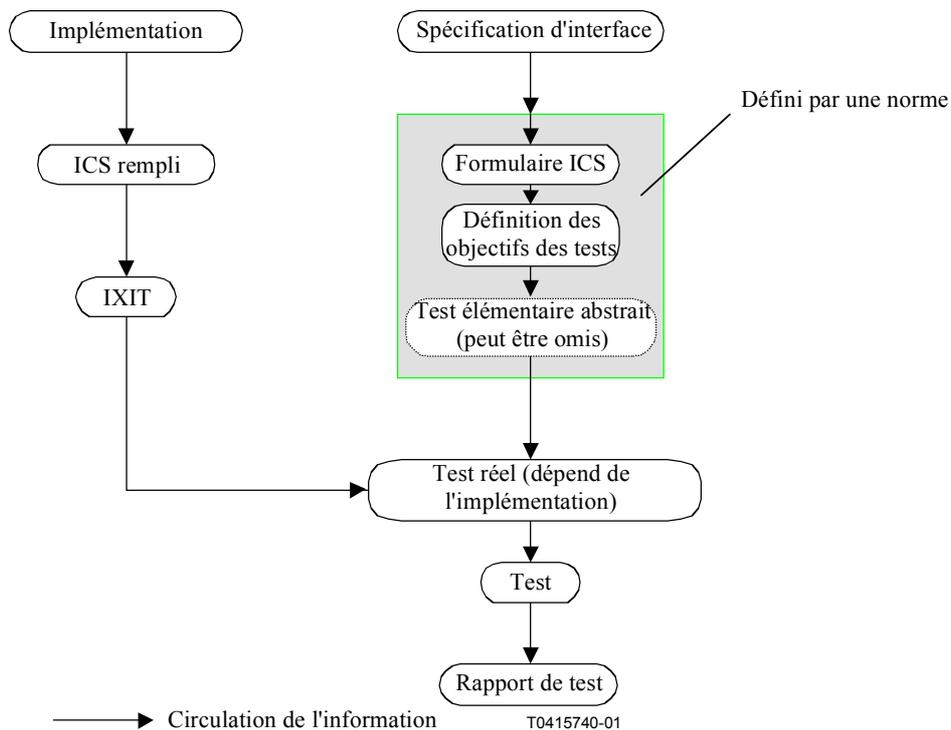


Figure 5-1/X.781 – Processus de test pour interfaces de type CORBA

6 Formulaire ICS pour interface de gestion de type CORBA

Le langage de définition d'interface (IDL) CORBA est utilisé pour définir des interfaces d'objets dans des systèmes de type CORBA. Le formulaire CIICS doit indiquer les caractéristiques IDL suivantes telles les modules IDL, les interfaces et d'autres types IDL. Ce paragraphe présente les caractéristiques du CORBA IDL et spécifie ensuite le formulaire CIICS conformément aux caractéristiques du langage.

6.1 Description générale de l'IDL

Le langage de définition d'interface (IDL) de l'OMG est le langage utilisé pour décrire les interfaces que les objets client appellent et que les implémentations d'objets fournissent. Une définition d'interface écrite en langage IDL OMG définit totalement l'interface et spécifie entièrement chacun des paramètres de fonctionnement. Une implémentation sous test (IUT, *implementation under test*) peut inclure une fonction client de l'interface ou une fonction serveur de l'interface. Ainsi, dans le formulaire d'interface, il doit y avoir un espace disponible pour spécifier l'aspect de la fonction (client ou serveur) implémentée par l'IUT.

La grammaire de l'IDL OMG ressemble beaucoup à la norme ANSI C++ proposée. Elle dispose de structures telles que des modules, des interfaces, des constantes, des types, des opérations, des attributs, des exceptions, des types de valeurs, des membres d'état et des déclarations de contexte. Conformément à la spécification de la syntaxe IDL, la relation de confinement entre ces structures peut être illustrée comme indiqué à la Figure 6-1.

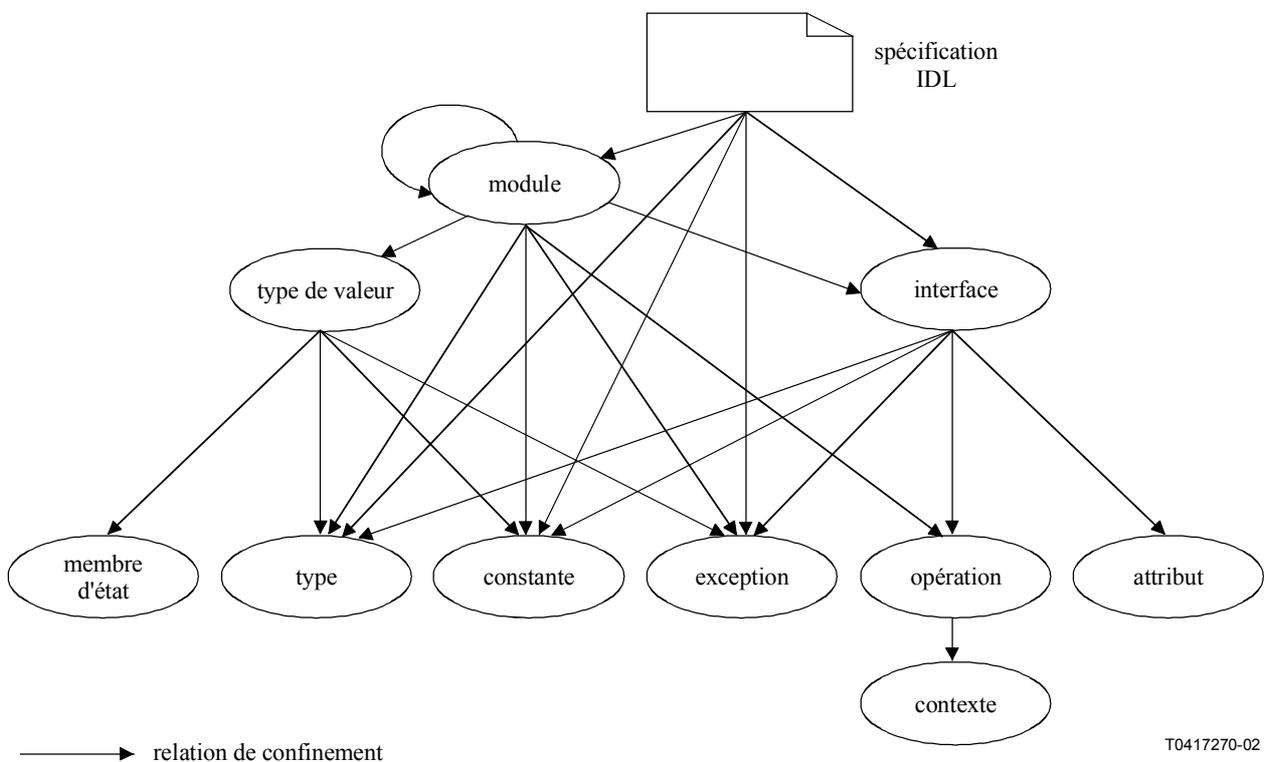


Figure 6-1/X.781 – Relation de confinement dans la spécification IDL

Les formulaires ICS doivent représenter la relation illustrée ci-dessus.

6.2 Prescriptions et directives de spécification des formulaires CIICS

Le style des spécifications de formulaire doit être conforme au style spécifié dans les paragraphes qui suivent. Les spécifications de formulaire doivent contenir les informations exigées dans la présente Recommandation. Des tableaux additionnels peuvent être inclus au besoin pour donner d'autres informations.

Il existe trois niveaux de documentation associés au formulaire CIICS, à savoir:

- les directives ou les outils définis dans les Recommandations pour la production des formulaires CIICS;

- b) un formulaire CIICS, associé à une norme relative à une gestion de réseau de type CORBA, qui doit être rempli par un fournisseur de l'implémentation et qui, lorsqu'il est rempli, constitue une déclaration CIICS;
- c) une déclaration CIICS préparée par le fournisseur de l'implémentation dans le cadre d'une déclaration de conformité à une norme relative à une gestion de réseau de type CORBA.

6.2.1 Instructions générales de la spécification des formulaires CIICS

La présente Recommandation spécifie des instructions nécessaires à l'élaboration d'une spécification de formulaire ICS de système de gestion de type CORBA. Les formulaires CIICS sont composés de quatre sous-formulaires: le formulaire de prise en charge de l'interface; le formulaire de prise en charge des attributs; le formulaire de prise en charge des opérations et le formulaire de prise en charge des types de données. Tous ces formulaires sont présentés sous forme de tableaux qui sont analogues à ceux du formulaire spécifié dans la Rec. UIT-T X.724.

Le § 6.2.2 décrit le formulaire CIICS pour les interfaces, le § 6.2.3 le formulaire CIICS pour les attributs, le § 6.2.4 le formulaire CIICS pour les opérations et le § 6.2.5 le formulaire pour les types de données complexes. Les Annexes A, B, C et D contiennent des exemples de spécification de formulaires CIICS, qui doivent être remplis par un fournisseur d'une implémentation.

Les notations communes suivantes, définies dans la Rec. UIT-T X.291 et l'ISO/CEI 9646-2 dans la Rec. UIT-T X.296 et l'ISO/CEI 9646-7 sont utilisées pour la colonne valeur "status" de la présente Recommandation:

- m Obligatoire
- o Optionnel
- Non applicable ou hors domaine d'application

NOTE – Les notations "m" et "o" sont préfixées par un "c:" lorsqu'elles sont intégrées à un élément optionnel du même tableau.

Les notations communes suivantes, définies dans la Rec. UIT-T X.291 et l'ISO/CEI 9646-2 et dans la Rec. UIT-T X.296 et l'ISO/CEI 9646-7 sont utilisées dans la colonne "support":

- Y Implémenté
- N Non implémenté
- Réponse non demandée
- Ig L'élément est ignoré (c'est-à-dire syntaxiquement mais pas sémantiquement)

La spécification du formulaire CIICS est formée en copiant les § 6.2.2, 6.2.3, 6.2.4 et 6.2.5 et en remplissant les tableaux à l'exception des colonnes "Support" et "Additional Information" et en étendant les tableaux restants pour respecter la spécification. Le formulaire CIICS doit contenir des tableaux pour tous les attributs, opérations, paramètres, exceptions et contextes définis dans le modèle d'information IDL, qu'ils soient dérivés ou non de superclasses ou ajoutés par redéfinition.

Pour former une déclaration CIICS à partir d'un formulaire CIICS, le fournisseur de la réalisation doit remplir la colonne "Support" et, si nécessaire, la colonne "Additional Information" de tous les tableaux du formulaire CIICS.

6.2.2 Formulaire de prise en charge d'interface

L'objet du formulaire pour les interfaces est d'avoir un mécanisme pour un fournisseur d'une réalisation qui déclare la conformité à une spécification d'interface de fournir l'information de conformité sous une forme normalisée.

Le formulaire de prise en charge de l'interface est analogue à celui présenté dans le Tableau 6-1:

Tableau 6-1/X.781 – Tableau de prise en charge d'interface

Index	Interface identifiant	Derived interface	Status	Support	Additional information

dans lequel:

- Le champ "Index" est constitué des numéros constitutifs pour les lecteurs de la déclaration CIICS pour désigner chaque élément. Chaque interface définie dans un modèle d'information IDL est donnée par un numéro unique au niveau de son indice.
- Le champ "Interface identifiant" est le nom absolu de l'interface qui se compose de l'espace nom et du nom d'interface.
- Le champ "Derived interface" est fourni pour l'interface ou les interfaces parents directes à partir desquelles l'interface considérée est dérivée, le cas échéant.
- Pour chaque interface instantiable, la colonne "Status" doit contenir "m" si elle est obligatoire, "o" si elle est optionnelle ou "-" si elle n'est pas instantiable ("- signifiant non applicable).
- Le champ "Additional Information" est appliqué pour indiquer de quel côté (client ou serveur) le système sous test est implémenté¹.

6.2.3 Formulaire de prise en charge d'attribut

L'objet du formulaire associé aux attributs est de permettre à un fournisseur d'une réalisation qui déclare la conformité à des spécifications d'attribut dans une interface, de disposer d'un mécanisme de déclaration par formulaire normalisé.

Ce formulaire de prise en charge d'attribut est analogue à celui du Tableau 6-2, qui doit être implémenté par interface IDL:

Tableau 6-2/X.781 – Tableau de prise en charge des attributs

Index	Attribute identifiant	Constraints and values	Get		Set		Additional information
			Status	Support	Status	Support	

Dans lequel:

- "Index" est la référence unique à l'intérieur d'une interface.
- L'"Attribute identifiant" est le nom relatif de l'attribut.
- Le champ "Constraints and values" est la description du type de données de l'attribut.

¹ Dans la plupart des cas l'implémentation sous test joue le rôle de serveur, mais lorsqu'elle sert à signaler une notification ou à autre chose, elle peut agir comme client et envoyer de l'information en invoquant une opération fournie par un autre serveur situé du côté testeur.

- Les champs "Get" et "Set" contiennent une partie "status" et "support". L'information "Status" de la colonne "Get" doit avoir toujours la valeur "m". Si l'attribut est un attribut normal, la valeur de l'information "Status" de "Set" doit être "m", tandis que si l'attribut est "Readonly", le contenu de la colonne "Status" doit être "-", indiquant que cela n'est pas applicable.
- Le champ "Additional Information" est de mettre à la disposition du réalisateur un espace pour ajouter des informations plus spécifiques concernant cet attribut, telles les valeurs par défaut lorsque l'objet est créé, ou le numéro d'indice de référence lorsque le type de cet attribut est "any". Voir le § 7.5 pour de plus amples détails.

Pour les attributs de type complexe, l'information détaillée relative au type est développée dans un tableau sous la rubrique "attribute support table", qui est appelé "data type support table" qui sera examiné au § 6.2.5.

6.2.4 Formulaire de prise en charge d'opérations

L'objet de ce formulaire relatif aux opérations est d'offrir un mécanisme pour le fournisseur d'une réalisation qui déclare la conformité à une spécification d'opération dans une interface pour fournir l'information de conformité sous forme normalisée.

Le formulaire de prise en charge d'opérations est constitué par les Tableaux 6-3 et 6-4, qui doivent être utilisés par interface IDL:

Tableau 6-3/X.781 – Tableau de prise en charge d'opérations (Partie 1)

Index	Operation identifiant	Status	Support	Additional information

Dans lequel:

Le champ "Operation identifiant" est le nom relatif de l'opération.

Le champ "Additional Information" a pour objet d'offrir un espace au réalisateur pour ajouter des informations plus spécifiques concernant l'opération.

Pour chaque opération, si cela est obligatoire, le contenu de la colonne "Status" doit être "m"; si des exceptions système (System Exception) sont utilisées, ce qui en général est spécifié non pas en IDL mais dans des déclarations de comportement associées, une explication doit être donnée dans le champ "Additional Information" pour spécifier le type d'exception de système utilisé; l'exception système NO_IMPLEMENT peut être utilisée pour indiquer que l'opération n'est pas prise en charge, le contenu de la colonne "Status" doit être "o"; s'il y a des exceptions d'utilisateur pour des paquetages conditionnels, le contenu de la colonne "Status" doit être "c".

D'autres colonnes ont une signification analogue tel que décrit au § 6.2.2.

Tableau 6-4/X.781 – Tableau de prise en charge d'opérations (Partie 2)

Index	Subindex	Operation field identifier	Constraints and values	Category	Field property	Status	Support	Additional information

Dans lequel:

- Le champ "Index" contient l'indice de chaque opération spécifiée dans le Tableau 6-1.
- Le champ "Subindex" sert à décrire la référence de chaque article associé à l'opération, laquelle est formée en reliant l'indice de l'opération avec un numéro unique au moyen d'un séparateur ".".
- Le champ "Operation field identifier" sert à décrire tous les identificateurs qui sont associés à l'opération, y compris les noms de paramètres, les identificateurs de contexte et les noms d'exception.
- Le champ "Constraints and values" est la description du type de données de chaque paramètre, la valeur retournée et l'exception définies dans cette opération. L'information détaillée sera étendue dans le "formulaire de prise en charge de type de données", comme cela est décrit au § 6.2.5.
- Le champ "Category" sert à distinguer les différents types de champ opération qui sont classés en deux types: "Parameter" et "termination", qui sont décrits plus loin au § 7.8.
- Le champ "Field property" est une autre description du champ "Category", qui décrit le mode ou les types détaillés du champ "Operation", comme cela est indiqué au § 7.9.
- Le champ "Additional Information" indique certaines informations spécifiques données par le réalisateur. Lorsque le type d'un paramètre est "any", un numéro d'indice de référence doit être indiqué dans ce champ. Voir le § 7.5 e) pour de plus amples détails.

6.2.5 Formulaire de prise en charge de type de données

Le formulaire de prise en charge de type de données est la description pour tous les types de données définis par l'utilisateur utilisés dans le modèle d'information IDL. Il s'agit d'une extension du "formulaire de prise en charge d'attribut" et du "formulaire de prise en charge d'opération". Le "tableau de prise en charge de type de données" est subdivisé en trois types de tableaux à savoir le "sous-tableau de prise en charge de paramètres", le "sous-tableau de prise en charge de valeur de renvoi" et le "sous-tableau de prise en charge d'exception" qui ont le même format que celui indiqué dans le Tableau 6-5.

Tableau 6-5/X.781 – Formulaire de prise en charge de type de données

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information

Dans lequel:

- "Index" est la référence à chaque identificateur de champ opération présent dans le Tableau 6-1.
- "Subindex" est la référence au champ enfant contenu dans le type super données.
- Le champ "Field identifier" décrit chaque nom de champ enfant d'un type de données construit.
- Le champ "Constraints and values" est la description du type de données de chaque champ enfant. Si le type de champ est un type complexe, il y aura un plus grand nombre de commentaires pour montrer ses champs enfant imbriqués sous la rangée qu'il occupe, ce processus sera continué jusqu'à ce que les types de données deviennent des types de base prédéfinis dans la spécification CORBA.
- Le champ "Additional Information" sert à spécifier certaines informations spécifiques que le réalisateur doit donner. Lorsque le type d'un champ est "quelconque", ce champ doit contenir un numéro d'indice de référence. Lorsque le champ est un membre d'état d'un type valuetype, sa publicité et son héritage doivent être décrits dans ce champ. Voir le § 7.5 e) pour de plus amples détails.

7 Instructions pour remplir le formulaire CIICS

Le présent paragraphe donne des instructions pour remplir chaque colonne définie au § 6.2 et ses paragraphes.

7.1 Définition du terme "Support"

Une capacité est dite prise en charge si la réalisation sous test (IUT) est en mesure de matérialiser la fonctionnalité spécifiée.

7.2 Colonne "Status"

Cette colonne indique le niveau de prise en charge requis pour la conformité avec une spécification IDL spécifique. Les valeurs sont celles définies au § 6.2.1.

Les directives pour remplir ce champ sont les suivantes:

- a) Dans le "tableau prise en charge d'interface", le champ "Status" doit contenir "m"; s'il est optionnel il doit contenir "o". Si une interface n'est pas instantiable, le statut doit contenir "-", signifiant non applicable.
- b) Dans le "tableau prise en charge d'attribut", la colonne "Status" doit contenir "m" pour toutes les rubriques "Get", et également pour la rubrique "Set" si l'attribut est normal; la rubrique "Set" doit contenir un "-" si l'attribut a le mot clé "readonly" qui le précède.

- c) Pour le "tableau prise en charge d'opération", ce champ doit contenir "m" si l'opération est nécessaire pour la fonctionnalité de gestion; dans les autres cas elle peut contenir "o". Si l'opération est obligatoire pour chaque champ "Operation" de cette opération, la colonne "Status" doit contenir "m", dans les autres cas elle doit contenir "c:m".
- d) Pour le "tableau prise en charge de type de données", ce n'est seulement lorsqu'un type est du type "union" que la colonne "Status" de ces champs peut contenir "o", étant donné que les unions sont principalement utilisées pour indiquer des options, en particulier lorsqu'il y a uniquement une seule dérivation avec "TRUE" comme valeur discriminante associée. Pour tous les autres types cette colonne doit contenir "m". Naturellement, lorsqu'un article contenant d'autres types est optionnel, tous les articles qu'il contient doivent ajouter "c:" comme préfixe pour ce champ, comme indiqué au § 6.2.1.

7.3 Colonne "prise en charge"

Cette colonne doit être remplie par le fournisseur ou le réalisateur pour indiquer le niveau d'implémentation de chaque article. La liste des indications que doit contenir ce champ est donnée au § 6.2.1.

On trouvera ci-après des directives sur la façon de remplir ce champ:

- a) Si un article est déclaré comme "supported", tous les articles obligatoires qu'il contient doivent être également pris en charge. Dans les autres cas, la colonne "support" peut juste être remplie par "N".
- b) Si la colonne "state" d'un article contient "-", la seule possibilité de contenu pour la colonne "support" correspondante est "-".
- c) Dans les tableaux de formulaire CIICS, chaque article marqué d'un "m" doit être pris en charge par l'implémentation sous test IUT.

7.4 Colonne "Constraints and Values"

Les colonnes "Constraints and Values" des tableaux (qui doivent être remplies dans la spécification de formulaire notifiée le cas échéant) contiennent les contraintes et les valeurs de l'article considéré (exemple, attribut, champ d'opération). Cette information peut inclure **le cas échéant**:

- a) toutes les contraintes concernant la prise en charge de l'article spécifique;
- b) les valeurs spécifiques pour l'attribut ou les paramètres d'opération qui sont pris en charge;
- c) les types autorisés conformément à la spécification normalisée.

Lorsque l'article décrit un type, le nom du type de cet article doit être rempli dans cette colonne. Cette règle est applicable aux tableaux suivants: "tableau de prise en charge d'attribut", "tableau de prise en charge d'opération" et "tableau de prise en charge de type de données".

Le nom de type utilisé pour remplir la colonne peut être décrit au moyen des expressions de génération suivantes:

```
<nom de type> ::= <nom de type de base> | <nom de type défini par l'utilisateur>
<nom de type de base> ::= void | short | unsigned short | long | unsigned long | long long |
unsigned long long | float | double | long double | char | boolean |
octet | string | TypeCode | objref | any | wchar | wstring | fixed
<nom de type défini par l'utilisateur> ::= chaîne décrivant un nom de type absolu
```

Pour chaque type de données défini par l'utilisateur, il est proposé d'indiquer dans cette colonne le "type kind" entre crochets.

"Type kind" est la description abstraite des types IDL. Le type de chaque type de base a le même nom que son nom de type. Les noms de type de tous les types définis par l'utilisateur sont: "struct", "union", "sequence", "array", "enum", "interface", "valuetype", "exception", "bstring"².

7.5 Colonne "Additional Information"

Cette colonne doit être remplie par le fournisseur pour donner certaines informations spécifiques sur l'implémentation non incluses dans les autres colonnes. On trouvera ci-après certaines règles générales qui doivent être suivies par les fournisseurs:

- a) Pour "le tableau de prise en charge d'interface", le rôle serveur ou client doit être explicitement spécifié dans ce champ. Lorsqu'il s'agit d'un rôle de serveur, "As server" doit être indiqué dans ce champ; dans les autres cas il faut indiquer "As client". Lorsque le rôle d'une interface est déterminé, toutes les opérations et attributs contenus dans cette interface joueront le même rôle.
- b) Pour le "tableau de prise en charge d'attribut", si un attribut a des valeurs par défaut lorsque l'objet est créé, il est proposé d'ajouter la valeur dans ce champ. Si l'attribut est à lecture seulement et que sa valeur initiale est donnée lorsque l'objet est créé, il faut indiquer "set by create". Si un attribut hérite d'une autre interface, il faut indiquer "Inherited".
- c) Pour le "tableau de prise en charge d'opération", si une opération est héritée d'une autre interface, il faut indiquer dans ce champ "Inherited". Si certains paramètres ont des valeurs par défaut, cela doit être indiqué dans la rangée correspondante de ce champ pour le paramètre considéré. Pour chaque exception définie par l'utilisateur, la condition d'existence de cette exception doit être précisée dans ce champ.
- d) Pour le "tableau de prise en charge d'attribut" et le "tableau de prise en charge d'opération", si un attribut ou une opération est surchargée dans une interface enfant, il faut indiquer dans ce champ "overload".
- e) Pour le "tableau de prise en charge d'attribut", le "tableau de prise en charge d'opération" et "tableau de prise en charge de type de données", lorsque le type d'un champ est "any", ce champ doit contenir une référence, indiquant le numéro d'indice d'un tableau appendice, dans lequel les types exécution à insérer dans cette indication "any" sont totalement décrits. Le numéro d'indice d'appendice recommandé est le même que l'indice de la première colonne de la même rangée, à ceci près qu'on ajoute un préfixe "A". Le "tableau de prise en charge d'appendice de quelconque" a le même format que le "tableau de prise en charge de type de données". (Pour de plus amples détails voir les exemples donnés dans l'Annexe C.)
- f) Pour le "tableau de prise en charge de type de données", lorsque le champ est le membre déclaratif d'un type de valeur, la publicité et l'héritage de ce membre déclaratif doivent être décrits. La publicité peut être "public state" ou "private state" et si ce membre déclaratif provient d'un héritage de son type de base, une phrase "inherited from <base valuetype name>" doit exister dans ce champ. L'Annexe D contient un exemplaire du formulaire CIICS pour des types de valeurs.
- g) Pour chaque opération si on utilise des exceptions système, le type des exceptions système utilisé doit être expliqué dans ce champ pour l'opération et si l'exception système NO_IMPLEMENT peut être utilisée pour spécifier "non pris en charge", la colonne "Status" correspondant doit contenir "o".

² "bstring" signifie "bounded string" (chaîne délimitée). En IDL, "bounded string" signifie une chaîne de longueur limitée. En CIICS, bstring <n> est utilisé pour décrire ce type dans lequel "n" est la longueur de la chaîne.

- h) Pour chaque tableau CIICS, si le fournisseur souhaite déclarer quelque chose de spécial il peut utiliser ce champ. Et si l'espace n'est pas suffisant, des tableaux élargis peuvent être ajoutés au formulaire CIICS.

7.6 Colonne "Index"

Chaque ligne du formulaire CIICS est numérotée sur sa gauche. Cette numérotation est un moyen d'identification unique de tous les détails d'implémentation possibles dans le formulaire CIICS.

Pour faire référence à des réponses individuelles on utilise la séquence suivante:

- a) une référence à la superclause de l'article;
- b) le caractère de séparation ".";
- c) un numéro unique.

Un exemple d'utilisation de cette notation est donné dans l'Annexe A.

La colonne "index" montre la relation de confinement entre des structures syntaxiques IDL. La relation de confinement est illustrée au § 6.1 "Description générale de l'IDL".

7.7 Colonne "Subindex"

Cette colonne a la même signification et le même format que la colonne "Index". Elle est également constituée de chiffres consécutifs.

- a) Pour le "Tableau de prise en charge d'opération", la superclause de "Subindex" est la colonne prépositive "Index" de la même rangée.
- b) Pour le "Tableau de prise en charge de type de données", la superclause de "Subindex" est l'indice de son article contenant direct, qui peut se trouver dans la colonne "Index" ou "Subindex".

7.8 Colonne "Category"

Cette colonne dans le "Formulaire de prise en charge d'opération" peut être remplie en utilisant deux types de choix, à savoir "parameter" ou "termination". Le contenu "parameter" indique que le "Operation field" est un paramètre d'invocation d'opération (incluant le contexte), tandis que le contenu "termination" indique que le "Operation field" est un type de situation de terminaison.

7.9 Colonne "Propriété du champ"

Cette colonne correspond à la colonne "Category". Si le contenu du champ "Category" est "parameter", cette colonne peut contenir l'information "in", "out", "inout" ou "context", indiquant le mode du paramètre. Si le contenu du champ "category" est "termination", le contenu de cette colonne peut être "success reply" ou "exception". Où "success reply" est la réponse souhaitée de l'opération, qui peut avoir une valeur de retour ou "void"; le terme "exception" indique que l'opération est interrompue pour une raison particulière.

7.10 Colonnes "Interface identifiant", "Attribute identifiant" et "Operation identifiant"

Ces colonnes doivent contenir les noms des interfaces, attributs ou opérations correspondantes.

NOTE – "Interface identifiant" est le nom absolu de l'interface, qui peut être identifié de manière unique dans l'espace nom du modèle total d'information IDL. "Attribute identifiant" et "Operation identifiant" sont uniquement des noms relatifs de l'article qui a l'espace nom uniquement dans son interface contenante.

7.11 Colonne "Derived interface"

La spécification IDL prend en charge la relation d'héritage. Cette colonne laisse un espace aux interfaces dérivées, le cas échéant.

NOTE – Seules les interfaces parents directes doivent être indiquées dans cette colonne; il n'est pas exigé d'indiquer les interfaces ancêtres.

ANNEXE A

Exemple de spécification de formulaire CIICS

A.1 Définition IDL CORBA

Dans la présente annexe, sont décrits les formulaires CIICS de "prise en charge d'interface", de "prise en charge d'attribut" et de "prise en charge d'opération". Nous donnons ci-après un exemple simple de définition IDL:

```
// DESCRIPTION:
//
// fichier source IDL pour l'exemple simple BANK
//
//*****
module BankModule {
//
// description simple d'un compte bancaire
//
interface account {
    bilan flottant d'attribut à lecture seulement;

    void makeLodgement (en f flottant);
    void makeWithdrawal (en f flottant);
};
//
// description simple d'un compte bancaire
//
interface currentAccount : account {
    readonly attribute float overdraftLimit;
};
//
// la banque constitue simplement les comptes
//
// bank:: reject apparaît si un nom de compte dupliqué est observé
//
interface bank {
    exception reject {};
    account newAccount (en nom chaîné) raises (rejet);
    currentAccount newCurrentAccount (en nom chaîné,
    en limite flottante)
        raises (rejet);
    void deleteAccount (in account a);
};
};
```

A.2 CIICS

D'après les définitions IDL ci-dessus, la déclaration CIICS pour ce simple modèle d'information IDL peut être illustré par les Tableaux A.1 à A.3:

Tableau A.1/X.781 – Prise en charge d'interface BankModule::bank

Index	Interface identifieur	Derived interface	Status	Support	Additional information
1	BankModule::bank		m	Y	As server

Tableau A.1.1/X.781 – Prise en charge d'attribut BankModule::bank

Il n'y a pas de définition d'attribut dans l'interface banque.

**Tableau A.1.2/X.781 – Prise en charge de l'opération BankModule::bank
(en qualité de serveur)**

Index	Operation identifieur	Status	Support	Additional information
1.2.1	newAccount	m	Y	
1.2.2	newCurrentAccount	m	Y	
1.2.3	deleteAccount	m	Y	

Index	Subindex	Operation field identifieur	Constraints and values	Category	Field property	Status	Support	Additional information
1.2.1	1.2.1.1	name	String	parameter	in	m	Y	
	1.2.1.2		BankModule::account (interface)	termination	success reply	m	Y	
	1.2.1.3	reason		termination	exception	m	Y	when the account object with the same name has existed
1.2.2	1.2.2.1	name	String	parameter	in	m	Y	
	1.2.2.2	limit	Float	parameter	in	m	Y	
	1.2.2.3		BankModule::currentAccount (interface)	termination	success reply	m	Y	
	1.2.2.4	reason		termination	exception	m	Y	when the current-Account object with the same name has existed
1.2.3	1.2.3.1	a	BankModule::account (interface)	parameter	in	m	Y	
	1.2.3.2		Void	termination	success reply	m	Y	

NOTE – Si le paramètre d'invocation prend la valeur par défaut, cela est illustré dans la colonne "Constraints and values".

Tableau A.2/X.781 – Prise en charge de l'interface BankModule::account

Index	Interface identifier	Derived interface	Status	Support	Additional information
2	BankModule::account		m	Y	As server

Table A.2.1/X.781 – BankModule::account attribute support

Index	Attribute identifier	Constraints and values	Get		Set		Additional information
			Status	Support	Status	Support	
2.1.1	balance	float	m	Y	-	-	default value is 0 when the object is created.

NOTE – Etant donné que l'attribut "balance" est un attribut "readonly", seul le "Get Status" est obligatoire, le "Set Status" n'est pas applicable.

Tableau A.2.2/X.781 – Prise en charge de l'opération BankModule::account (assurant les fonctions de serveur)

Index	Operation identifier	Status	Support	Additional information
2.2.1	makeLodgement	m	Y	
2.2.2	makeWithdrawal	m	Y	

Index	Subindex	Operation field identifier	Constraints and values	Category	Field property	Status	Support	Additional information
2.2.1	2.2.1.1	f	float	parameter	in	m	Y	
	2.2.1.2		void	termination	success reply	m	Y	
2.2.2	2.2.2.1	f	float	parameter	in	m	Y	
	2.2.2.2		void	terminator	success reply	m	Y	

Tableau A.3/X.781 – Prise en charge de l'interface BankModule::currentAccount

Index	Interface identifier	Derived interface	Status	Support	Additional information
3	BankModule::current Account	BankModule::account	m	Y	As server

Tableau A.3.1/X.781 – Prise en charge de l'attribut BankModule::currentAccount

Index	Attribute identifier	Constraints and values	Get		Set		Additional information
			Status	Support	Status	Support	
3.1.1	balance	float	m	Y	-	-	Inherited
3.1.2	overdraft Limit	float	m	Y	-	-	the value is set by create

NOTE – Etant donné que l'attribut "balance" est hérité de l'interface BankModule::account, on indique "Inherited" dans la colonne "Additional Information".

Si un attribut ou une opération est surchargé dans une interface enfant, il faut indiquer "overload" dans le champ "Additional information".

Tableau A.3.2/X.781 – Prise en charge de l'opération BankModule::currentAccount (rôle de serveur)

Index	Operation identifier	Status	Support	Additional information
3.2.1	makeLogement	m	Y	Inherited
3.2.2	makeWithdrawal	m	Y	Inherited

Index	Subindex	Operation field identifier	Constraints and values	Category	Field property	Status	Support	Additional information
3.2.1	3.2.1.1	f	float	parameter	in	m	Y	
	3.2.1.2		void	termination	success reply	m	Y	
3.2.2	3.2.2.1	f	float	parameter	in	m	Y	
	3.2.2.2		void	termination	success reply	m	Y	

Ce qui précède est un exemple très simple qui montre la spécification du formulaire CIICS pour le "formulaire de prise en charge d'interface", le "formulaire de prise en charge d'attribut" et le "formulaire de prise en charge d'opération". L'Annexe B contiendra un exemple illustrant le "formulaire de prise en charge de type de données".

ANNEXE B

Exemple de spécification de "formulaire de prise en charge de type de données"

B.1 Définition IDL CORBA

La présente annexe montre un exemple de formulaire contenant plusieurs types de données complexes IDL définis par l'utilisateur qui peuvent être utilisés pour illustrer le format du "formulaire de prise en charge de type de données". La définition IDL est la suivante:

```
module typeExample {

    typedef short shortArray[2][3];
    typedef sequence<short,4> shortSequence;

    enum Color { red, blue, green};
    exception ExceptionType {
        short    number;
        string   reason;
    };

    typedef short short_alias;

    struct DataStructure {
        short_alias  s;
        float        f;
    };

    union UN_DS {
        short s;
        DataStructure ds;
    };

    interface typeInterface {
        UN_DS op (
            in      shortArray      p1_sA,
            in      shortSequence    p2_sS,
            inout   Color            p3_color,
            out     DataStructure     p4_UT
        )
        raises (ExceptionType)
        context ( "key1", "key2" ) ;
    };
};
```

B.2 CIICS

Conformément à la définition IDL ci-dessus, le formulaire CIICS pour ce modèle d'information donné en exemple peut être illustré dans les tableaux suivants:

Tableau B.1/X.781 – Prise en charge d'interface typeExample::typeInterface

Index	Interface identifiant	Derived interface	Status	Support	Additional information
1	TypeExample::typeInterface		m	Y	As server

Tableau B.1.1/X.781 – Prise en charge d'attribut typeExample::typeInterface

L'interface typeExample::typeInterface n'a pas d'attribut défini.

Tableau B.1.2/X.781 – Prise en charge d'opération typeExample::typeInterface

Index	Operation identifier	Status	Support	Additional information
1.2.1	Op	m	Y	

Index	Subindex	Operation field identifier	Constraints and values	Category	Field property	Status	Support	Additional information
1.2.1	1.2.1.1	p1_sA	::typeExample::shortArray (array[2][3])	parameter	in	m	Y	
	1.2.1.2	p2_sS	::typeExample::shortSequence (sequence<4>)	parameter	in	m	Y	
	1.2.1.3	p3_color	::typeExample::Color (enum)	parameter	inout	m	Y	
	1.2.1.4	p4_UT	::typeExample::DataStructure (struct)	parameter	out	m	Y	
	1.2.1.5		key1	parameter	context	m	Y	
	1.2.1.6		key2	parameter	context	m	Y	
	1.2.1.7		::typeExample::UN_DS (union)	termination	success reply	m	Y	
	1.2.1.8	Exception Type	::typeExample::ExceptionType (exception)	termination	exception	m	Y

Sous-tableau de prise en charge de paramètre

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information
1.2.1.1	1.2.1.1.1		Short	m	Y	
1.2.1.2	1.2.1.2.1		Short	m	Y	
1.2.1.4	1.2.1.4.1	s	::typeExample::short_alias (short)	m	Y	
	1.2.1.4.2	f	Float	m	Y	

NOTE – Le champ enfant d'un type union doit contenir au moins un article comme champ enfant obligatoire. On suppose ici que le champ "s" est obligatoire et que le champ "ds" est optionnel. En réalité, le caractère obligatoire ou optionnel d'un champ enfant du type "union" dépend de la sémantique de l'implémentation.

Sous-tableau de prise en charge de la valeur retournée

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information
1.2.1.7	1.2.1.7.1	S	Short	m	Y	
	1.2.1.7.2	Ds	::typeExample::DataStructure (struct)	o	Y	
	1.2.1.7.2.1	S	::typeExample::short_alias (short)	c:m	Y	
	1.2.1.7.2.2	F	Float	c:m	Y	

NOTE – Le champ enfant d'un type union doit contenir au moins un article comme champ enfant obligatoire. Ici, le champ "f" est supposé être obligatoire et le champ "ds" facultatif. Ainsi le "Status" de tous les éléments contenus dans "ds" est entièrement préfixé par "c:", signifiant que le champ enfant des éléments "s" et "f" ne peut être obligatoire que lorsque "ds" est pris en charge.

Sous-tableau de prise en charge d'exception

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information
1.2.1.8	1.2.1.8.1	number	short	m	Y	
	1.2.1.8.2	reason	string	m	Y	

ANNEXE C

Exemple de spécification de "formulaire de prise en charge d'appendice de type quelconque"

C.1 Définition CORBA IDL

La présente annexe présente un exemple contenant une interface qui a un attribut de type "quelconque" et une opération avec un paramètre tout-type, qui peut être utilisé pour montrer le format d'un "formulaire de prise en charge d'appendice de type quelconque". La définition IDL est la suivante:

```
Module anyExample {
    struct DataStructure {
        short    s;
        string   str;
    };

    interface anyInterface {
        attribute any anyAttr;
        void op ( inout any p1 ) ;
    };
};
```

C.2 CIICS

D'après la définition IDL ci-dessus, la déclaration CIICS pour ce modèle d'information donné en exemple peut être illustré par les tableaux suivants:

Tableau C.1/X.781 – Prise en charge de l'interface anyExample::anyInterface

Index	Interface identifieur	Derived interface	Status	Support	Additional information
1	anyExample::anyInterface		m	Y	As server

Tableau C.1.1/X.781 – Prise en charge de l'attribut anyExample::anyInterface attribute support

Index	Attribute identifieur	Constraints and values	Get		Set		Additional information
			Status	Support	Status	Support	
1.1.1	anyAttr	any	m	Y	m	Y	See appendix Table C.1.1.1

NOTE – Comme le type de l'attribut "anyAttr" est "any", le champ "Additional Information" contient le numéro d'indice de référence de l'appendice du Tableau d'appendice correspondant: "C.1.1.1", qui se compose d'un "A" suivi par l'indice de cet attribut "1.1.1". On suppose ici que le type de temps de traitement possible autorisé pour cet attribut est "unsigned short", "long" et "float".

Tableau C.1.2/X.781 – Prise en charge de l'opération anyExample::anyInterface

Index	Operation identifieur	Status	Support	Additional information
1.2.1	op	m	Y	As Server

Index	Subindex	Operation field identifieur	Constraints and values	Category	Field property	Status	Support	Additional information
1.2.1	1.2.1.1	p1	any	parameter	inout	m	Y	See appendix Table C.1.2.1.1
	1.2.1.2		void	termination	Success reply	m	Y	

NOTE – Comme le type du paramètre "p1" est "any", le champ "Additional Information" pour cet article contient le numéro d'indice de référence de l'appendice du Tableau de l'appendice correspondant: "C.1.2.1.1", qui se compose d'un "A" suivi du sous-indice de ce paramètre "1.2.1.1", on suppose ici que le type de temps de traitement possible autorisé pour ce paramètre est "string", et "anyExample::DataStructure".

Les tableaux d'appendice pour les types "any" décrits précédemment sont les suivants:

Tableau C.1.2.1/X.781 – <::anyExample::anyInterface::anyAttr→anyAttr any types support>

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information
1	1.1		long	m	Y	
2	2.1		ushort	m	Y	
3	3.1		float	m	Y	

Tableau C.1.2.1.1/X.781 – <::anyExample::anyInterface::op→p1 any types support>

Index	Subindex	Field identifier	Constraints and values	Status	Support	Additional information
1	1.1		string	m	Y	
2	2.1		anyExample::dataStructure (struct)	m	Y	
	2.1.1	s	short	m	Y	
	2.1.2	str	string	m	Y	

NOTE – Le "tableau de prise en charge appendice de tous types" a le même format que le "tableau de prise en charge de type de données" et le contenu de ce tableau est constitué par tous les types autorisés qui peuvent être insérés dans ce champ.

ANNEXE D

Exemple de spécification du "formulaire de prise en charge de valuetype"

D.1 Définition CORBA IDL

La présente annexe montre un exemple contenant une définition de valuetype avec héritage et est défini comme paramètre d'une opération d'une autre interface qui peut être utilisée pour décrire le "formulaire de prise en charge de valuetype". La définition IDL est la suivante:

```
Module valueModule {
    valuetype BaseValue {
        public short id;
    };
    valuetype ValueExample : BaseValue {
        private string name;
        public long value;

        factory init(in string name, in long value);
        //method
        void local_op ();
    };

    interface aInterface {
        void remote_op (in ValueExample p1);
    };
};
```

D.2 Déclaration CIICS

D'après la définition IDL ci-dessus, la déclaration CIICS dans le cas du modèle d'informations données en exemple peut être illustrée dans les tableaux suivants.

Tableau D.1/X.781 – Prise en charge de l'interface valueModule::aInterface

Index	Interface identifiant	Derived interface	Status	Support	Additional information
1	valueModule::aInterface		m	Y	As server

Tableau D.1.1/X.781 – Prise en charge de l'attribut valueModule::aInterface

L'interface valueModule::ValueInterface n'a pas d'attribut défini.

Tableau D.1.2/X.781 – Prise en charge de l'opération valueModule::aInterface

Index	Operation identifiant	Status	Support	Additional information
1.2.1	remote_op	m	Y	As Server

Index	Subindex	Operation field identifiant	Constraints and values	Category	Field property	Status	Support	Additional information
1.2.1	1.2.1.1	p1	ValueModule::ValueExample (valuetype)	parameter	in	m	Y	
	1.2.1.2		void	termination	Success reply	m	Y	

Sous-tableau de prise en charge des paramètres

Index	Subindex	Field identifiant	Constraints and values	Status	Support	Additional information
1.2.1.1	1.2.1.1.1	Id	short	m	Y	Public state, inherited from valueModule::BaseValue
1.2.1.2	1.2.1.2.1	Name	string	m	Y	Private state
	1.2.1.2.2	Value	long	m	Y	Public state

NOTE – D'après les tableaux ci-dessus, on constate que lorsqu'il est transféré sous forme de paramètre, le type valuetype est principalement traité de la même façon que struct, sauf que valuetype peut contenir des états hérités de ces valuetype de base. Etant donné que "id" est un membre de "public state" dont valueModule::ValueExample hérite de valueModule::BaseValue, l'état "public state, hérité de valueModule::BaseValue" est indiqué dans le champ "Additional Information". Etant donné que "name" est un membre d'état privé de valueModule::valueExample, "private state" et contenu dans le champ "Additional Information".

Bien qu'il y ait des opérations d'initialisateur "init()" et des opérations ordinaires ou local_op() définies dans cette valuetype valueModule::ValueExample, elles n'apparaissent pas dans le formulaire ICS, bien que ces types d'opérations soient des opérations locales, qui sont en dehors du domaine d'application des déclarations de conformité d'implémentation d'interfaces.

SÉRIES DES RECOMMANDATIONS UIT-T

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