



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.600

(11/2000)

SÉRIE Z: LANGAGES ET ASPECTS GÉNÉRAUX
LOGICIELS DES SYSTÈMES DE
TÉLÉCOMMUNICATION

**Architecture de l'environnement de traitement
réparti**

Recommandation UIT-T Z.600

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Z
LANGAGES ET ASPECTS GÉNÉRAUX LOGICIELS DES SYSTÈMES DE TÉLÉCOMMUNICATION

TECHNIQUES DE DESCRIPTION FORMELLE	
Langage de description et de spécification (SDL)	Z.100–Z.109
Application des techniques de description formelle	Z.110–Z.119
Diagrammes des séquences de messages	Z.120–Z.129
LANGAGES DE PROGRAMMATION	
CHILL: le langage de programmation de l'UIT-T	Z.200–Z.209
LANGAGE HOMME-MACHINE	
Principes généraux	Z.300–Z.309
Syntaxe de base et procédures de dialogue	Z.310–Z.319
LHM étendu pour terminaux à écrans de visualisation	Z.320–Z.329
Spécification de l'interface homme-machine	Z.330–Z.399
QUALITÉ DES LOGICIELS DE TÉLÉCOMMUNICATION	Z.400–Z.499
MÉTHODES DE VALIDATION ET D'ESSAI	Z.500–Z.599

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

Architecture de l'environnement de traitement réparti

Résumé

La présente Recommandation décrit l'architecture de l'environnement de traitement réparti (DPE) qui représente l'environnement d'exécution pour des services et des applications de télécommunications et informatiques.

L'architecture DPE a pour objet de mettre en évidence des besoins techniques détaillés conduisant à des spécifications qui fournissent une aide aux fournisseurs DPE dans l'élaboration de leurs produits et aux réalisateurs d'application dans la compréhension de l'infrastructure fournie par l'environnement DPE.

La présente Recommandation se fonde sur les travaux de l'équipe noyau TINA dans le cadre du consortium TINA ainsi que sur plusieurs projets annexes menés à bien par les sociétés membres du consortium TINA-C, en appui de l'équipe noyau.

La présente Recommandation traite des points suivants:

- explication de la relation entre les concepts de modélisation, dans la mesure où une telle relation joue un rôle dans l'architecture de traitement;
- description du réseau de transport noyau (KTN) qui joue, dans l'environnement DPE, un rôle analogue à celui du système de signalisation dans les télécommunications;
- cadre général d'interfonctionnement pour l'environnement DPE;
- expression des besoins concernant les services noyau de l'environnement DPE.

Il est possible de mettre en évidence un certain nombre de services d'ingénierie DPE prenant en charge l'exécution de services de télécommunication. Ces services objet de l'environnement DPE sont identifiés et associés aux fonctions et aux transparences du modèle de référence pour le traitement réparti ouvert (RM-ODP). L'expression détaillée des besoins et les spécifications des services objet de l'environnement DPE appellent une étude ultérieure.

Il est nécessaire de pouvoir gérer l'environnement DPE, les services objet de l'environnement DPE et les applications mises en place dans un environnement DPE. Le contenu et les modalités de cette gestion appellent également une étude ultérieure.

Certaines applications ne nécessitent pas la totalité des services noyau DPE identifiés par la présente Recommandation. Ceci conduit à la définition de profils DPE prenant en charge divers types de services et d'applications. Il est nécessaire de spécifier les services noyau qui sont obligatoires pour un profil donné. La définition et l'utilisation des profils DPE appellent une étude ultérieure. L'environnement DPE doit prendre en charge la qualité de service (QS) nécessaire aux services et aux applications. Les modalités de cette prise en charge appellent une étude ultérieure.

Source

La Recommandation Z.600 de l'UIT-T, élaborée par la Commission d'études 10 (2001-2004) de l'UIT-T, a été approuvée le 24 novembre 2000 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
1.1	Objet..... 2
1.2	Objectif 2
1.3	Autres fonctionnalités 2
2	Références normatives 2
2.1	Recommandations Normes internationales identiques 2
2.2	Autres Recommandations 3
2.3	Autres spécifications 3
3	Définitions 3
3.1	Termes définis par les fondements du modèle de référence ODP 3
3.2	Termes définis par l'architecture du modèle de référence ODP 3
3.3	Définitions relatives à l'architecture de l'environnement de traitement réparti 3
4	Abréviations..... 5
5	Introduction..... 6
5.1	Architecture DPE..... 7
5.2	Points de référence DPE 7
5.3	Extensions de l'architecture CORBA..... 8
6	Concepts de modélisation de l'ingénierie..... 9
6.1	Cycle de vie des objets..... 10
6.2	Communication au sein de l'environnement DPE 10
6.2.1	Communication par opérations..... 11
6.2.2	Communication par flux 11
6.3	Mappage de la vue informatique vers la vue d'ingénierie 12
6.4	Liaisons 12
6.4.1	Liaison par opérations ou par flux..... 12
6.4.2	Liaisons implicites ou explicites..... 12
6.5	Canaux 12
6.6	Constitution des liaisons 14
6.6.1	Introduction 14
6.6.2	Interfaces de flux programmables 14
6.6.3	Commande et gestion de flux de média..... 15
7	Réseau de transport noyau 16
7.1	Couche de messagerie..... 17
7.2	Couche Transport..... 17

	Page	
7.3	Prescriptions pour l'environnement DPE.....	17
8	Interfonctionnement DPE	18
8.1	Cadre d'interfonctionnement général	19
8.2	Extensions DPE de l'interfonctionnement CORBA 2.....	19
9	Services d'ingénierie fournis par le noyau DPE.....	20
9.1	Introduction.....	21
9.2	Caractéristiques concernant le temps réel.....	21
9.3	Prescriptions concernant le modèle d'exécution	22
9.4	Prescriptions de modularité.....	22
9.4.1	Souplesse de l'architecture DPE	22
9.4.2	Prise en charge de protocoles multiples.....	23
9.4.3	Schéma de communication générique	23
9.4.4	Prise en charge de liaisons souples.....	24
9.5	Besoins fonctionnels	24
9.5.1	Prise en charge des interfaces de flux.....	24
9.5.2	Prise en charge de l'exécution simultanée	25
9.5.3	Mappage souple des événements vers les fils d'exécution	25
9.5.4	Gestion de la simultanéité.....	26
9.5.5	Processus générique d'ordonnancement	27
9.5.6	Service de temps	27
9.6	Besoins non fonctionnels	27
9.6.1	Degré de finesse des objets.....	27
9.6.2	Empreinte mémoire de taille réduite	27
9.6.3	Documentation du comportement temporel	28
9.7	Interfaces multiples.....	28
10	Services objet.....	28
10.1	Introduction.....	28
11	Bibliographie	30
Annexe A	– Concepts de modélisation d'entreprise – Cadre général pour la propagation de prescriptions dans un marché de télécommunications ouvert.....	31
A.1	Domaine d'application	31
A.2	Termes et définitions	31
A.3	Concepts de modélisation commerciale	32
A.3.1	Cadre général.....	32
A.3.2	Segmentation des points de référence	33
A.3.3	Combinaison de rôles commerciaux en domaines administratifs commerciaux	35
A.3.4	Délégation.....	36

	Page
Annexe B – Concepts de modélisation d'ingénierie.....	37
B.1 Concepts d'ingénierie de base.....	37
Appendice I – Scénario du cycle de vie d'un objet	40
Appendice II – Liaison flexible.....	40
Appendice III – Problèmes de prise en charge d'interfaces de flux	41

Architecture de l'environnement de traitement réparti

1 Domaine d'application

La présente Recommandation décrit l'environnement de traitement réparti (DPE, *distributed processing environment*) prenant en charge l'exécution d'applications de télécommunications réparties. L'environnement DPE peut être considéré comme l'infrastructure sur laquelle peuvent s'exécuter des applications de télécommunications réparties telles que des applications multimédia et en temps réel.

Elle contient l'expression des besoins et la définition des fonctionnalités de base de l'environnement DPE prenant en charge l'exécution d'applications de télécommunications réparties. Toute application de télécommunications répartie conçue conformément au modèle RM-ODP tirera des bénéfices de l'utilisation de la plate-forme fournie par l'environnement DPE. L'environnement DPE proprement dit se fonde sur les concepts et les principes du modèle RM-ODP ([2] et [3]).

Le modèle RM-ODP spécifie un **point de vue** (concernant un système) sous la forme d'une abstraction élaborée en utilisant un ensemble sélectionné de concepts d'architecture et de règles de structuration permettant de se concentrer sur diverses préoccupations au sein d'un système. Le modèle RM-ODP fournit les points de vue suivants:

- point de vue entreprise;
- point de vue informationnel;
- point de vue informatique;
- point de vue d'ingénierie;
- point de vue technologique.

La présente Recommandation se concentre sur le point de vue d'ingénierie du modèle RM-ODP. Elle constitue une spécialisation et un affinement du langage d'ingénierie du modèle RM-ODP permettant la prise en compte des besoins propres au domaine des télécommunications.

Le modèle RM-ODP spécifie une **transparence vis-à-vis de la répartition** qui est la propriété permettant de masquer, pour un utilisateur donné, le comportement potentiel de certaines parties d'un système réparti. Le modèle RM-ODP fournit les transparences suivantes vis-à-vis de la répartition:

- transparence vis-à-vis de l'accès;
- transparence vis-à-vis des défaillances;
- transparence vis-à-vis de l'emplacement;
- transparence vis-à-vis de la migration;
- transparence vis-à-vis du transfert;
- transparence vis-à-vis de la réplication;
- transparence vis-à-vis de la persistance;
- transparence vis-à-vis des transactions.

Le modèle RM-ODP spécifie des **fonctions** pour la prise en charge du traitement réparti ouvert. Le modèle RM-ODP fournit les fonctions suivantes:

- fonctions de gestion;
- fonctions de coordination;
- fonctions de référentiel;
- fonctions de sécurité.

La transparence du modèle RM-ODP vis-à-vis de la répartition et les fonctions de traitement ODP sont associées aux services noyau et objet de l'environnement DPE. La présente Recommandation se base sur l'utilisation des fonctions d'ingénierie, avec un affinement ou une spécialisation éventuelle pour l'environnement DPE.

L'environnement DPE adopte l'architecture CORBA 2 du groupe OMG comme la technologie primaire de base, de sorte que les services objet CORBA communs sont incorporés dans l'environnement DPE lorsque cela est approprié.

La présente Recommandation encourage en particulier l'étude du document traitant de l'architecture CORBA en temps réel [6], étant donné que cette spécification du groupe OMG satisfait dès à présent un certain nombre de besoins identifiés dans la présente Recommandation.

1.1 Objet

L'architecture DPE a pour objet de mettre en évidence des besoins techniques détaillés conduisant à des spécifications qui aident les fournisseurs DPE dans l'élaboration de leurs produits et les réalisateurs d'application dans la compréhension de l'infrastructure fournie par l'environnement DPE. Les besoins énoncés se basent sur la validation des résultats de projets d'implémentation.

La présente Recommandation contient, en résumé:

- les besoins techniques détaillés;
- les énoncés des fonctionnalités requises.

1.2 Objectif

La présente Recommandation a pour objectif de fournir une description abstraite de l'environnement DPE. Les implémentations de l'environnement DPE doivent prendre en charge des applications se conformant au modèle RM-ODP, c'est-à-dire élaborées conformément aux concepts et à la méthodologie de ce modèle. La description abstraite de l'environnement DPE est indépendante de l'implémentation et doit être utilisée par des fournisseurs de plates-formes DPE comme référence pour la construction de plates-formes se conformant à l'environnement DPE.

1.3 Autres fonctionnalités

La présente Recommandation décrit des services dont on estime qu'ils seront présents dans un grand nombre d'implémentations de l'environnement DPE. Il décrit ces services sans indiquer de quelle manière ils doivent être combinés.

La présente Recommandation n'impose aucune restriction pour une implémentation concrète de l'environnement DPE.

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.

2.1 Recommandations | Normes internationales identiques

- [1] UIT-T X.901 (1997) | ISO/CEI 10746-1:1998, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: aperçu général.*

- [2] UIT-T X.902 (1995) | ISO/CEI 10746-2:1996, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: fondements.*
- [3] UIT-T X.903 (1995) | ISO/CEI 10746-3:1996, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: architecture.*
- [4] UIT-T X.931 (1999) | ISO/CEI 14752:2000, *Technologies de l'information – Traitement réparti ouvert – Prise en charge des protocoles pour les interactions informatiques.*

2.2 Autres Recommandations

- [5] UIT-T Z.130 (1999), *Langage de définition d'objet de l'UIT.*

2.3 Autres spécifications

- [6] OMG Document formal/01.09.34, *The Common Object Request Broker: Architecture and Specification*, Revision 2.5 (*Courtier de demande d'objet commun: architecture et spécification*).

3 Définitions

Les définitions suivantes s'appliquent aux fins de la présente Recommandation.

3.1 Termes définis par les fondements du modèle de référence ODP

La présente Recommandation utilise les termes suivants définis dans la Rec. UIT-T X.902:

- a) architecture (d'un système);
- b) domaine;
- c) interface;
- d) objet;
- e) squelette (ou gabarit);
- f) fil d'exécution.

3.2 Termes définis par l'architecture du modèle de référence ODP

La présente Recommandation utilise les termes suivants définis dans la Rec. UIT-T X.903:

- a) liant (ou objet lieur);
- b) capsule;
- c) canal;
- d) point de reprise;
- e) grappe;
- f) noyau;
- g) nœud;
- h) souche (ou talon).

3.3 Définitions relatives à l'architecture de l'environnement de traitement réparti

La présente Recommandation utilise les termes suivants:

3.3.1 domaine administratif commercial: un domaine administratif commercial est défini par les besoins d'un ou de plusieurs rôles commerciaux et régi par un objectif commercial unique.

- 3.3.2 rôle commercial:** fonction dont on estime qu'elle sera exécutée par un compétiteur dans un environnement commercial de télécommunications.
- 3.3.3 interface API de communication:** interface de programmation d'application fournissant l'accès aux services de communication.
- 3.3.4 environnement de traitement réparti (DPE, *distributed processing environment*):** l'environnement DPE est l'infrastructure qui fournit l'environnement d'exécution, incluant la transparence vis-à-vis de la répartition, pour des applications réparties au sein d'un système. L'environnement DPE fournit à ses utilisateurs une vue transparente vis-à-vis de la répartition. Les utilisateurs de l'environnement DPE sont les concepteurs et les réalisateurs d'applications et de services. L'environnement DPE se constitue d'un ensemble de nœuds DPE interconnectés.
- 3.3.5 interface API DPE:** interface de programmation d'application fournissant l'accès aux services de l'environnement DPE.
- 3.3.6 point de référence DPE:** vue d'ingénierie d'un point de référence.
- 3.3.7 objet informatique d'ingénierie (eCO, *engineering computational object*):** un objet eCO est la représentation (correspondance biunivoque), du point de vue de l'ingénierie, d'un objet informatique qui encapsule l'état, les données et les traitements.
- 3.3.8 services d'ingénierie:** les services d'ingénierie prennent en charge la transparence de l'environnement DPE vis-à-vis de la répartition. Il existe un certain nombre de services d'ingénierie (par exemple, un référentiel ou un adaptateur d'objet) qui sont nécessaires pour un grand nombre d'applications et jouent un rôle fondamental dans la construction d'applications s'exécutant sur l'environnement DPE.
- 3.3.9 protocole général entre courtiers ORB (GIOP, *general inter-ORB protocol*):** le protocole GIOP est un protocole de messagerie utilisé pour la communication d'objets entre divers nœuds DPE; il est spécifié dans la référence [6].
- 3.3.10 protocole Internet entre courtiers ORB (IIOP, *Internet inter-ORB protocol*):** le protocole IIOP se constitue du mappage du protocole GIOP vers le protocole TCP/IP; il est spécifié dans la référence [6].
- 3.3.11 point de référence entre domaines:** identique à un point de référence. Un point de référence entre domaines est présent de manière implicite entre deux domaines, sauf indication contraire dans la présente Recommandation.
- 3.3.12 point de référence intradomanial:** point de référence non visible à l'extérieur d'un domaine administratif commercial. Un point de référence intradomanial est utilisé principalement à des fins d'acquisition.
- 3.3.13 réseau de transport noyau (KTN, *kernel transport network*):** le réseau de transport noyau est le réseau qui transporte des invocations et des réponses entre les nœuds DPE.
- 3.3.14 environnement informatique et de communication natif (NCE, *native computing and communication environment*):** l'environnement NCCE est un ordinateur, constitué en général d'un ensemble de ressources (matériel et système d'exploitation) qui fournissent conjointement une plateforme donnée. L'environnement NCCE est en dehors du domaine d'application de l'environnement DPE. Chaque nœud DPE s'exécute sur un environnement NCCE dont il utilise les ressources et les services. Un nœud DPE utilise, pour la communication avec d'autres nœuds DPE, une pile de protocole TCP/IP fournie par l'environnement NCCE.
- 3.3.15 courtier de demande d'objet (ORB, *object request broker*):** le courtier de demande d'objet fournit un mécanisme de communication transparente pour des demandes issues d'objets client répartis à destination des implémentations d'objet serveur et le renvoi des réponses de l'objet serveur vers l'objet client demandeur. Le courtier de demande d'objet est spécifié dans la référence [6].
- 3.3.16 services objet:** les services objet prennent en charge l'exécution d'applications réparties.

3.3.17 point de référence: un point de référence marque la frontière entre deux domaines administratifs commerciaux. Un point de référence se constitue d'un ensemble de plusieurs spécifications, liées à des points de vue au sein d'un contexte, qui définissent des contraintes de fonctionnement pour un ou plusieurs points de référence (se référer à l'Annexe A).

4 Abréviations

La présente Recommandation utilise les abréviations suivantes.

AAL	couche d'adaptation ATM (<i>ATM adaptation layer</i>)
ACID	atomicité, cohérence, isolation et durabilité (<i>atomicity, consistency, isolation and durability</i>)
API	interface de programmation d'application (<i>application programming interface</i>)
CMIP/CMIS	protocole commun de transfert des informations de gestion/service commun de transfert d'informations de gestion (<i>common management information protocol/common management information service</i>)
CORBA	architecture de courtier de demande d'objet commun (<i>common object request broker architecture</i>)
DCE-CIOP	environnement de calcul réparti – protocole commun entre courtiers ORB (<i>distributed computing environment – common inter-ORB protocol</i>)
DPE	environnement de traitement réparti (<i>distributed processing environment</i>)
eCO	objet informatique d'ingénierie (<i>engineering computational object</i>)
ESIOP	protocole entre courtiers ORB spécifique à l'environnement (<i>environment specific inter-ORB protocol</i>)
GIF	cadre d'interfonctionnement général (X.931) (<i>general interworking framework</i>);
GIOP	protocole général entre courtiers ORB (<i>general inter-ORB protocol</i>)
IDL	langage de définition d'interface (<i>interface definition language</i>)
IIOP	protocole Internet entre courtiers ORB (<i>Internet inter-ORB protocol</i>)
IOR	référence d'objet interface (<i>interface object reference</i>)
IPC	communication entre processus (<i>interprocess communications</i>)
KTN	réseau de transport noyau (<i>kernel transport network</i>)
LAN	réseau local (<i>local area network</i>)
MPEG	groupe d'experts pour les images animées (<i>moving picture experts group</i>)
NCCE	environnement informatique et de communication natif (<i>native computing and communication environment</i>)
ODL	langage de définition d'objet (<i>object definition language</i>)
OMG	groupe de gestion d'objets (<i>object management group</i>)
ORB	courtier de demande d'objet (<i>object request broker</i>)
POSIX	interface portable de système d'exploitation (<i>portable operating system interface</i>)
QoS	qualité de service
RM-ODP	modèle de référence du traitement réparti ouvert (<i>reference model for open distributed processing</i>)

SCCP	sous-système commande de connexions sémaphores (<i>signalling connection control part</i>)
SNMP	protocole simple de gestion de réseau (<i>simple network management protocol</i>)
SS7	système de signalisation 7
TCAP	sous-système application pour la gestion des transactions (<i>transaction capabilities application part</i>)
TCP/IP	protocole de commande de transmission/protocole Internet (<i>transmission control protocol/Internet protocol</i>)
TINA	architecture d'intégration en réseau des informations de télécommunication (<i>telecommunication information networking architecture</i>)
UDP	protocole datagramme d'utilisateur (<i>user datagram protocol</i>)

5 Introduction

La présente Recommandation définit l'architecture et les prescriptions permettant d'édifier des environnements de traitement réparti importants prenant en charge l'exécution de services de télécommunication et d'applications, y compris la prise en charge de services et d'applications en temps réel.

On peut définir un ensemble de composants logiciels appelé "middleware" prenant en charge l'interconnexion de systèmes répartis et de services. Les composants middleware sont des produits logiciels pouvant être positionnés entre ou sous des applications et au-dessus du logiciel système dans les systèmes informatiques, les réseaux et les services de communication. Dans la plupart des cas, la couche middleware masque l'hétérogénéité des systèmes sous-jacents, à savoir les langages de programmation, les systèmes informatiques et les protocoles réseaux. La couche middleware fournit, du point de vue du réalisateur du système, une vue homogène d'un environnement hétérogène réparti structuré correctement, au sein de laquelle les applications et les services interagissent par le biais d'interfaces normalisées ouvertes. L'environnement DPE est une instance de middleware.

L'environnement DPE est une infrastructure fournissant l'environnement d'exécution, y compris la transparence vis-à-vis de la répartition, pour des applications réparties au sein d'un système. L'environnement DPE fournit à ses utilisateurs une vue transparente vis-à-vis de la répartition. Les utilisateurs sont les concepteurs et les réalisateurs d'applications et de services. L'environnement DPE se constitue d'un ensemble de nœuds DPE interconnectés.

La présente Recommandation fournit une description abstraite de l'environnement DPE. Les implémentations de l'environnement DPE doivent prendre en charge des applications se conformant au modèle RM-ODP, c'est-à-dire élaborées conformément aux concepts et à la méthodologie de ce modèle. La description abstraite de l'environnement DPE est indépendante de l'implémentation et doit être utilisée par des fournisseurs de plates-formes DPE comme référence pour la construction de telles plates-formes.

L'environnement DPE contient un noyau DPE qui fournit la prise en charge de la gestion du cycle de vie des objets et de la communication entre objets. La gestion du cycle de vie des objets englobe les capacités de création et de suppression des objets en cours d'exécution. La communication entre objets fournit les mécanismes de prise en charge de l'invocation des opérations offertes par les interfaces d'opérations d'objets distants. Le noyau de l'environnement DPE fournit des fonctions de base, indépendantes de la technologie, qui correspondent aux capacités de la plupart des systèmes d'exploitation, à savoir l'exécution d'applications et la communication entre applications.

5.1 Architecture DPE

L'architecture DPE, représentée par la Figure 1, constitue l'entité fédératrice qui englobe, dans une implémentation normalisée, la totalité des fonctionnalités obligatoires décrites par la présente Recommandation. Elle fournit à cet effet, pour un environnement hétérogène réparti, une vue homogène et structurée correctement au sein de laquelle les applications et les services interagissent par le biais d'interfaces normalisées ouvertes. Cette architecture masque l'hétérogénéité des systèmes sous-jacents, à savoir les langages de programmation, les systèmes informatiques et les protocoles réseaux.

L'architecture DPE fournit la couche middleware permettant l'interaction d'objets informatiques d'ingénierie situés sur des nœuds DPE distants. Elle fournit également des outils de diagnostic et de configuration destinés aux administrateurs de réseau.

On peut classer les composants de l'architecture DPE selon les catégories suivantes:

- noyau DPE;
- services objet DPE;
- outils de prise en charge DPE.

Un nœud DPE est géré par un noyau DPE unique (voir également l'Annexe B).

Un point de référence DPE peut être présent entre deux nœuds DPE quelconques.

5.2 Points de référence DPE

Les concepts de **domaine administratif commercial** et de **point de référence** sont utilisés comme défini dans l'Annexe A.

Un point de référence représente un ensemble de spécifications apparentées, situé entre deux domaines administratifs commerciaux. Le point de référence de l'environnement DPE est la vue d'ingénierie d'un point de référence qui prend en charge le point de référence apparenté au service, comme indiqué dans la Figure 2. Les services et les applications utilisent l'interface API de l'environnement DPE pour accéder aux fonctionnalités de l'environnement DPE.

Un point de référence marque une frontière entre deux domaines administratifs commerciaux; un point de référence DPE marque de ce fait une frontière de domaine entre deux nœuds DPE actifs au sein de deux domaines administratifs commerciaux distincts.

La communication entre nœuds DPE se fait par le biais d'un point de référence DPE.

La communication entre services ou entre nœuds DPE peut se faire de l'une des manières suivantes:

- **Client-serveur**: ce mode classique de communication entre objets se fonde sur une séparation du logiciel entre tâches de serveur et tâches de client. Un client émet à destination d'un serveur, en utilisant un protocole donné, des demandes concernant des informations ou une action et le serveur répond à ces demandes. Ce type d'interaction peut mettre en jeu un serveur centralisé unique ou un ensemble de serveurs. Ce modèle permet d'implanter des clients et des serveurs de manière indépendante sur des nœuds d'un réseau; il peut utiliser éventuellement des logiciels et des systèmes d'exploitation adéquats, par exemple un serveur rapide et des clients peu onéreux.
- **Transfert de messages**: ce mode de communication met en jeu des processus parallèles. Un service s'exécutant sur un nœud donné peut émettre un message destiné à un service s'exécutant sur le même nœud ou sur un nœud différent. L'environnement d'exécution traite la transmission effective du message. Un environnement avec transfert de messages fournit des primitives d'émission et de réception de messages, ces primitives pouvant être synchrones, asynchrones ou les deux. Une émission synchrone reste en attente jusqu'au moment de la réception du message par le processus récepteur (l'émetteur reste bloqué entre temps). Ceci permet à l'émetteur de savoir si le message a été reçu correctement ou non. Une

émission asynchrone se borne à placer le message dans une file d'attente d'émission sans attendre la réception de message. Une primitive de réception synchrone restera en attente tant qu'il n'y a pas de message à lire, alors qu'une primitive de réception asynchrone se terminera immédiatement en renvoyant soit un message reçu, soit une indication d'absence de message. Les messages peuvent être destinés à un service nommé ou à une boîte aux lettres nommée pouvant être lue par un ou plusieurs services. Les messages peuvent posséder, ou non, un type. Ils peuvent également posséder une priorité permettant de lire en premier les messages de priorité la plus élevée.

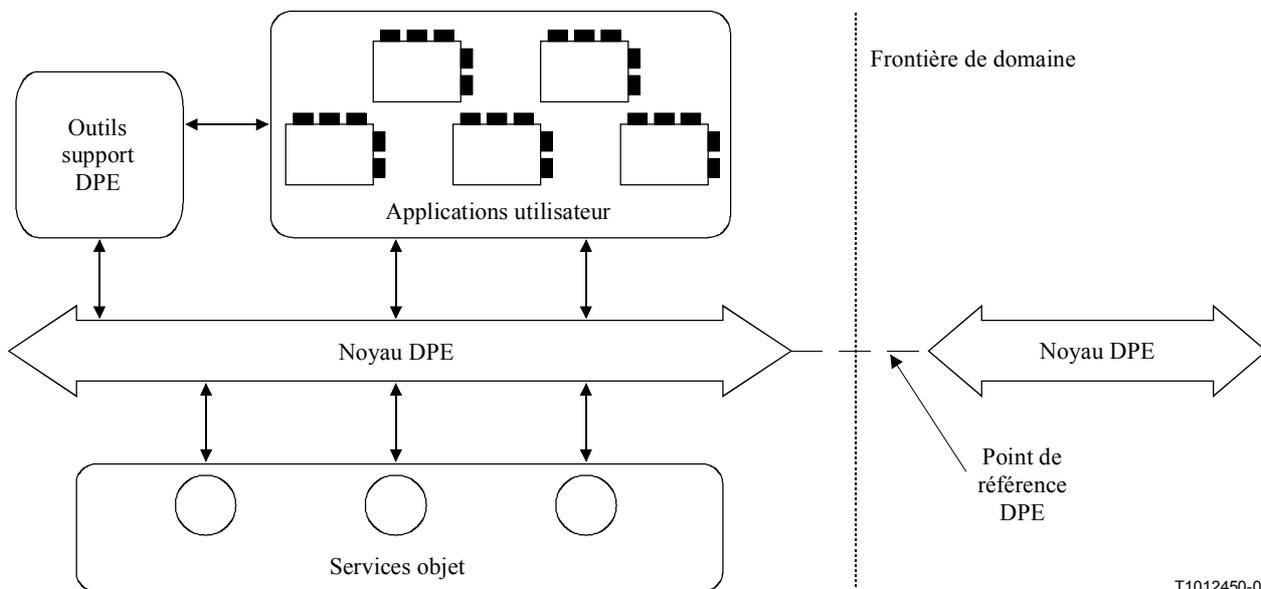
5.3 Extensions de l'architecture CORBA

L'architecture CORBA 2 du groupe OMG [6] est un mappage de l'architecture DPE vers une technologie spécifique. Bien qu'elle soit basée en grande partie sur l'architecture CORBA 2, l'architecture DPE n'en duplique pas les fonctionnalités et décrit uniquement les prescriptions supplémentaires qui sont du ressort du domaine des télécommunications.

Le noyau DPE nécessite des fonctionnalités supplémentaires par rapport à celles qui sont fournies par l'architecture CORBA 2 du groupe OMG. Les fonctionnalités requises sont, par exemple, les suivantes:

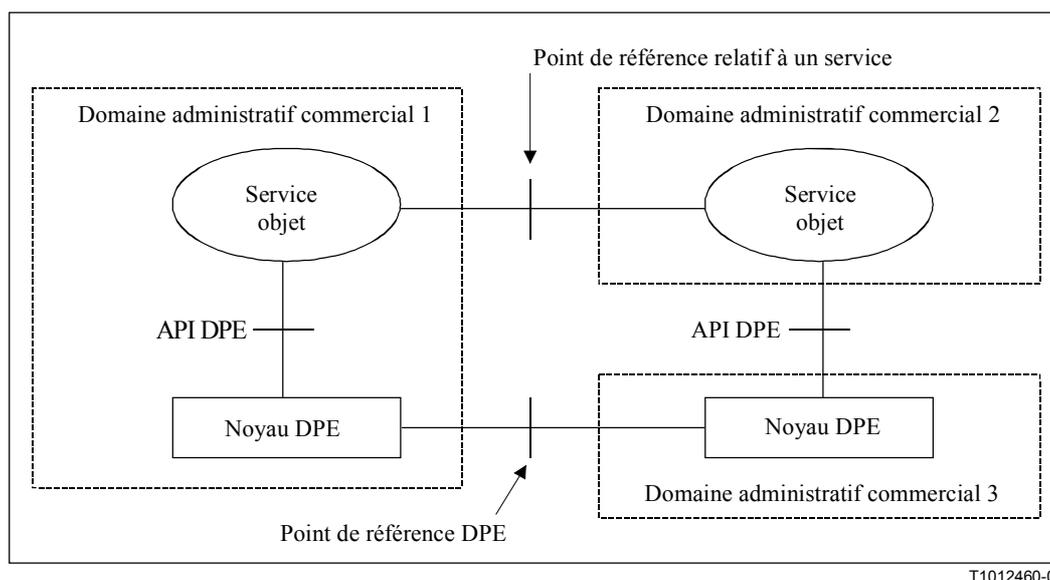
- prise en charge d'un modèle objet étendu, par exemple pour des interfaces multiples et des interfaces de flux;
- prise en charge et intégration de flux de média;
- intégration des caractéristiques de qualité de service;
- prise en charge de services et d'applications en temps réel;
- prise en charge de divers réseaux pour les communications entre environnements DPE.

La construction d'un environnement DPE peut, au niveau des objets "service", réutiliser de nombreux services "objet" OMG communs, comme spécifié pour les services CORBA par le groupe OMG [7].



T1012450-00

Figure 1/Z.600 – Architecture



T1012460-00

Figure 2/Z.600 – Points de référence et interfaces API

6 Concepts de modélisation de l'ingénierie

La présente Recommandation utilise les concepts de base de modélisation de l'ingénierie tels qu'ils sont définis en [3] et décrits dans l'Annexe B. L'Annexe B traite en particulier des notions suivantes:

- nœud DPE;
- noyau DPE;
- capsule;
- grappe;

- canal;
- cycle de vie d'objet;
- communication au sein de l'environnement DPE:
 - communication par opérations;
 - communication par flux;
- liaisons.

6.1 Cycle de vie des objets

Un certain nombre d'étapes a été mis en évidence pour le cycle de vie des objets d'ingénierie. La Figure 3 représente les étapes suivantes qui sont significatives pour l'environnement DPE:

- **mise en place:** le code exécutable est mis en place dans un ou plusieurs nœuds sous la forme de squelettes d'objet eCO, ce qui permet au nœud DPE de créer une instance d'objet eCO pour laquelle il dispose d'un code exécutable;
- **création:** une instance d'objet eCO est créée au sein d'une grappe;
- **activation:** l'activation d'un objet le fait passer dans un état actif, ce qui signifie que des ressources informatiques sont allouées à l'instance de l'objet et que cette dernière est prête à recevoir des invocations et à y répondre;
- **désactivation:** la désactivation fait passer un objet dans un état non actif, ce qui signifie que ses ressources informatiques sont libérées et que ses interfaces ne sont plus en mesure de recevoir des invocations et d'y répondre. La désactivation d'un objet implique également le stockage d'un point de reprise pouvant être utilisé pour une réactivation ultérieure de l'objet;
- **suppression:** une instance d'objet eCO est supprimée dans une grappe;
- **retrait:** le squelette d'objet eCO est retiré du nœud.

L'Appendice I décrit un scénario classique de cycle de vie d'objet.

CYCLE DE VIE D'OBJET	mise en place		
	INSTANCE DE CYCLE DE VIE	création	
		EXÉCUTION	activation
			désactivation
		suppression	
	retrait		

Figure 3/Z.600 – Etapes du cycle de vie d'un objet

6.2 Communication au sein de l'environnement DPE

Ce paragraphe décrit les concepts de modélisation informatique pour les communications entre objets au sein d'un environnement DPE.

Les interfaces informatiques constituent le seul moyen d'accès aux services offerts par un objet informatique à son environnement. Un objet informatique interagit avec d'autres objets informatiques en invoquant les opérations informatiques (appelées par la suite "opérations") offertes par ces autres objets ou en échangeant des flux avec ces derniers. On distingue les interfaces d'opération et les interfaces de flux.

6.2.1 Communication par opérations

Les interactions au niveau d'une interface d'opération sont structurées sous la forme d'invocations d'une ou de plusieurs opérations et de réponses à ces invocations. Les opérations sont classées en distinguant:

- les interrogations;
- les annonces.

Contrairement à ce qui se passe pour une interaction par interrogation, une interaction par annonce ne donne pas lieu au renvoi d'un résultat du serveur vers le client et ce dernier n'est pas informé du résultat positif ou négatif de l'invocation. Un message d'opération se constitue d'un ensemble de données de longueur finie.

6.2.1.1 Référence d'interface

Les objets informatiques définissent des interfaces informatiques, caractérisées par des types d'interface et des attributs de service, qui constituent des points d'interaction mis à disposition d'autres objets. Les interfaces d'ingénierie des objets eCO correspondent aux interfaces des objets informatiques concernés.

Les interfaces des objets eCO sont décrites par des références d'interface d'ingénierie qui spécifient les informations nécessaires pour identifier sans ambiguïté un site d'ingénierie d'une interface et pour raccorder cette dernière.

Les références d'interface d'ingénierie peuvent être transmises à travers des nœuds DPE hétérogènes et faire l'objet d'une comparaison d'égalité (par exemple pour activer une identification d'interface après la migration d'une instance d'objet).

6.2.2 Communication par flux

Une interface de flux est une entité abstraite qui représente un point d'extrémité de communication pouvant jouer l'un des rôles suivants:

- source de certains flux d'informations;
- puits pour certains autres flux d'informations; ou
- simultanément les rôles de source et de puits pour des flux d'informations.

Lorsque des objets interagissent par le biais d'interfaces de flux, l'échange d'informations se fait sous la forme d'écoulements de flux entre objets. Chaque écoulement de flux se constitue d'une succession unidirectionnelle de bits avec une structure de trame (format et codage de données) et des paramètres de qualité de service. Un flux de message peut être constitué d'informations de longueur finie.

Si l'on prend comme référence l'écoulement du flux, l'objet qui est la source du flux est appelé le producteur et l'objet qui est le puits est appelé le consommateur. Un objet qui offre une interface de flux spécifie les écoulements de flux qui se produisent au niveau de cette interface ainsi que, pour chaque flux, si l'objet en est le producteur ou le consommateur. Pour plus de détails, voir la Rec. UIT-T Z.130.

NOTE – Les flux contenant des objets appellent une étude ultérieure qui dépend de la définition d'un objet dans la norme MPEG.

6.3 Mappage de la vue informatique vers la vue d'ingénierie

Voir l'Annexe B.

6.4 Liaisons

Conformément au modèle RM-ODP, une liaison offre un contexte contractuel résultant d'un comportement d'établissement donné. Pour que deux objets puissent communiquer par le biais de leurs interfaces, l'environnement d'exécution (c'est-à-dire, l'environnement DPE) doit fournir les mécanismes permettant de les lier. Le § 6.6 et l'Appendice II contiennent des développements sur les liaisons.

6.4.1 Liaison par opérations ou par flux

On peut, en fonction des types d'interfaces liées, classer les liaisons de la manière suivante:

- liaison par opérations;
- liaison par flux.

6.4.2 Liaisons implicites ou explicites

On peut en outre classer les liaisons de la manière suivante, en fonction du type du comportement d'établissement:

- liaison explicite résultant d'actions explicites de liaison sur les objets qui feront partie de la liaison;
- liaison implicite effectuée par un participant externe, c'est-à-dire lorsque l'utilisateur n'effectue pas d'action de liaison explicite.

Voir Figure 4.

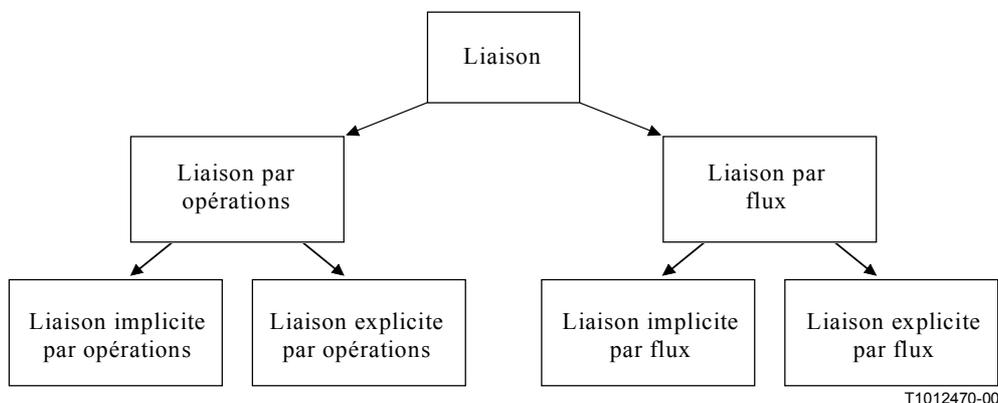


Figure 4/Z.600 – Relations de liaison

6.5 Canaux

Le canal est une notion ODP utilisée pour la prise en charge des liaisons explicites. La fonctionnalité d'ingénierie se rapportant à un canal résulte des besoins de prise en charge d'interactions d'objets informatiques d'une manière indépendante de leur répartition. On définit les trois services de canal suivants:

- le service de souche;
- le service de liant;
- le service objet protocole.

La description de ces services peut se faire en les regroupant sous la forme de services étroitement liés du côté client et du côté serveur. Il s'ensuit qu'un canal se présente d'une manière symétrique comme décrit dans la Figure 5.

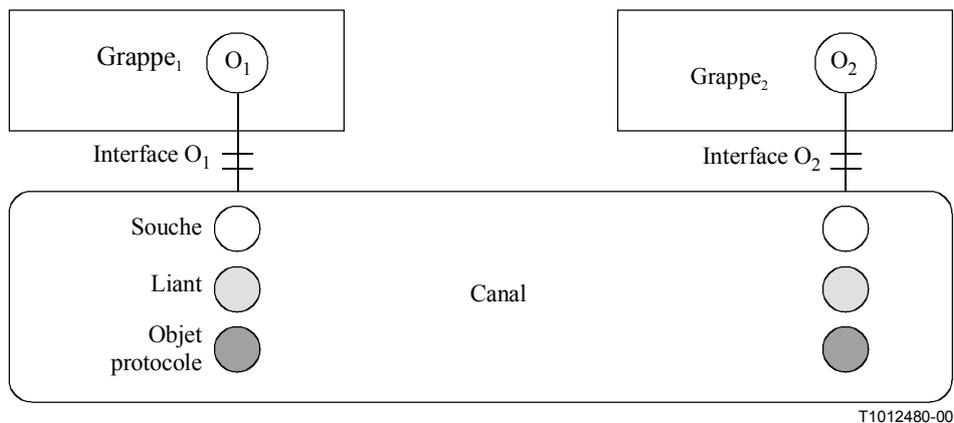


Figure 5/Z.600 – Souche, liant et objet protocole au sein d'un canal

Les **souches** prennent en charge les interfaces du canal vers la représentation d'ingénierie des objets informatiques qui interagissent. Dans le cas d'une communication par opérations, les souches fournissent des interactions supplémentaires et/ou des informations d'interaction supplémentaires entre les objets informatiques. Les souches exécutent des procédures préliminaires et finales avant et après l'invocation effective de l'opération. Elles fournissent une fonctionnalité d'adaptation dans la mesure où elles effectuent les transformations nécessaires à la fourniture de fonctionnalités de transparence telles que la transparence vis-à-vis de l'accès et la transparence vis-à-vis des transactions. Les souches fournissent également des fonctionnalités de compression et de décompression dans le cas d'une communication par flux.

Les **liants** associés aux points d'extrémité des canaux des objets eCO maintiennent l'intégrité de la liaison au cours de l'interaction entre ces objets. Ils gèrent des informations concernant le canal, telles que le contexte de l'association ou un tampon de données. Ils sont également responsables de la validation de la référence de l'interface et de l'interaction avec un service de relocalisation ou tout autre service auxiliaire, afin de conserver des informations cohérentes au sujet de l'emplacement de l'interface dans le cas d'une erreur de liaison (suite à une défaillance, une migration, une désactivation, etc.).

Les services de souche et de liant décrits précédemment fournissent principalement la transparence vis-à-vis de l'accès et de la localisation. La fourniture d'autres transparences peut nécessiter que les souches et les liants disposent de fonctionnalités supplémentaires, correspondant par exemple à des services de relocalisation et de gestion de transaction.

Les objets "**protocole**" permettent l'interaction distante entre les objets informatiques concernés au sein d'un canal. Les objets protocole assurent la communication entre nœuds par le biais d'un protocole qui garantit le respect de la sémantique de l'interaction. Les objets "protocole" fournissent la fonctionnalité de transparence vis-à-vis de l'accès utilisé par l'objet "souche", qui a connaissance d'une interface informatique spécifique qui lui est liée.

NOTE 1 – Les interactions au sein d'un nœud DPE peuvent éventuellement s'effectuer en dehors de la prise en charge de services de protocole.

NOTE 2 – Le terme "distant" indique, dans ce contexte, la traversée de frontières entre nœuds.

6.6 Constitution des liaisons

6.6.1 Introduction

La liaison implicite d'interfaces informatiques au sein de l'environnement DPE est prise en charge par le protocole GIOP, ce qui est obligatoire pour tous les environnements DPE. L'objet "client" connaît dans ce cas uniquement la référence de l'objet "interface" (IOR) lorsqu'il invoque des opérations sur cette interface sans disposer de la commande de la liaison. La spécification de l'architecture CORBA 2 [6] décrit de manière approfondie les modalités d'implémentation des liaisons implicites.

Pour que deux objets puissent interagir par le biais d'un écoulement de flux, il est nécessaire que chacun d'eux fournisse une interface de flux et que ces deux interfaces soient liées.

La différence essentielle entre les interactions par le biais d'interfaces d'opérations et les interactions par le biais d'interfaces de flux est que les premières possèdent une structure d'invocation d'opérations et de réponses à ces invocations, alors qu'aucune structure de ce type n'est imposée pour les secondes. Un ensemble de flux d'informations est établi entre deux objets, avec des paramètres spécifiques de qualité de service, une fois qu'une liaison a été établie entre les interfaces de flux de ces objets. Le producteur du flux procède par la suite à l'insertion d'informations dans le flux que le consommateur extrait du flux en vue de les utiliser. Aucune interaction explicite n'a lieu entre le producteur et le consommateur au cours de cet échange d'informations. La gestion de ces flux s'effectue par le biais des interfaces d'opérations. La liaison entre le producteur et le consommateur sera interrompue ultérieurement par le producteur, le consommateur ou par un objet tiers. Il s'ensuit que le processus d'interaction par le biais d'interfaces de flux se constitue d'un transfert asynchrone de messages, alors que le processus d'interaction par le biais d'interfaces d'opérations peut correspondre à une invocation de procédure à distance ou à un transfert asynchrone de messages.

Le canal de flux utilise une structure d'adaptateur souche-liant-protocole comparable à celle du canal d'opérations.

La prise en charge de la liaison de flux dans l'environnement DPE peut s'effectuer de l'une des manières suivantes:

- interfaces de flux programmables;
- commande et gestion d'interfaces de flux.

6.6.2 Interfaces de flux programmables

Les prescriptions pour la prise en charge des interfaces de flux au sein de l'environnement DPE englobent les suivantes:

- capacité de définition et de référence pour des interfaces de flux avec une sécurité concernant le type;
- capacité de liaison de plusieurs interfaces de flux au moyen de divers objets "liaison de flux" avec, dans le cas idéal, certaines contraintes de qualité de service;
- capacité de programmation des interfaces de flux;
- capacité de synchronisation de plusieurs flux.

L'observation de ces prescriptions fournit un environnement DPE qui offre une prise en charge intrinsèque permettant d'effectuer des actions de création, de référence et de programmation sur des interfaces de flux. Un adaptateur d'objet "flux multimédia" fournit les services suivants (de manière directe ou par délégation):

- liaison explicite (avec une sécurité concernant le type) d'interfaces de flux multimédia, avec la capacité de spécifier des contraintes de qualité de service;
- configuration, réservation et contrôle d'admission pour des ressources;

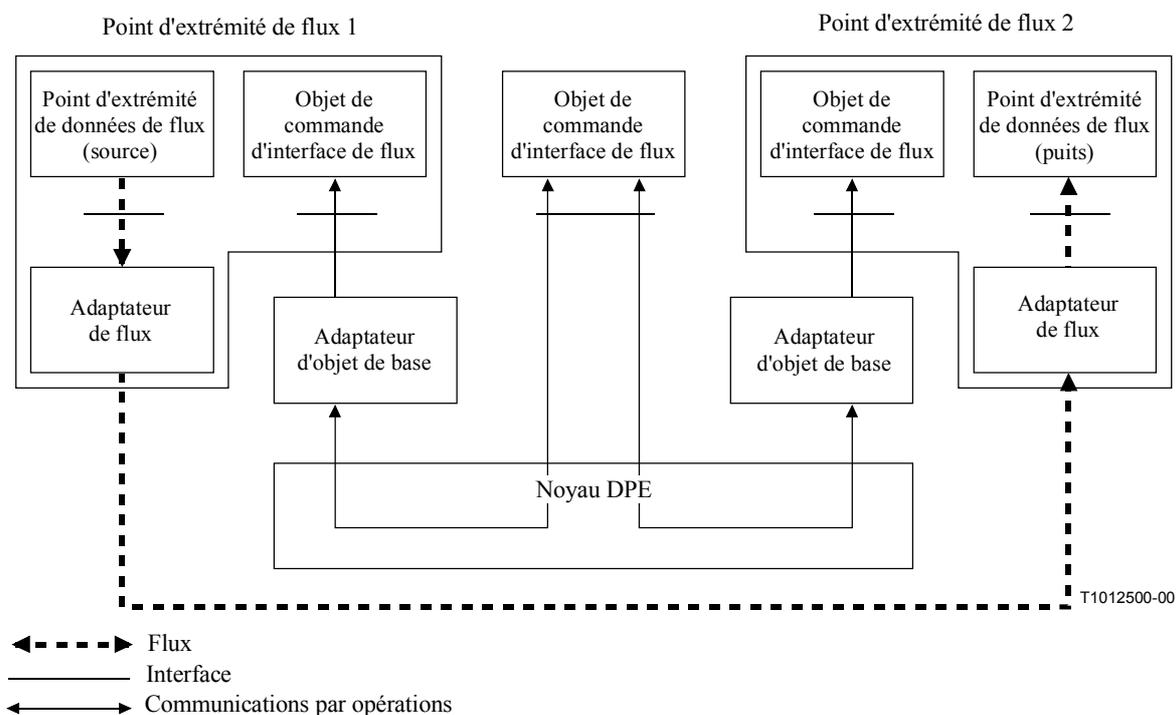


Figure 7/Z.600 – Architecture de flux OMG

L'objet "commande d'interface de flux" utilise un adaptateur d'objet qui émet et reçoit des messages de commande en conformité avec l'architecture CORBA 2.

Il n'est pas nécessaire que les objets "source" et "puits" et l'adaptateur de flux soient visibles comme des entités distinctes lorsque la terminaison d'un flux se fait au niveau matériel.

La spécification OMG-A/V adoptée pour la commande et la gestion de flux définit les composants constitutifs d'un flux et une interface pour des objets de commande et de gestion de flux, ainsi que des commandes d'interface de flux associées aux points d'extrémité de flux individuels.

Les références d'interface de l'architecture CORBA 2 pour des objets de commande d'interface de flux sont utilisées en particulier pour faire référence à tous les points d'extrémité de flux dans des paramètres d'opérations de commande de flux définies par la spécification. Il en résulte que, dans le cas de la commande et de la gestion de flux de média, la spécification n'impose pas la définition d'un nouveau type de données IDL pour les références d'interface de flux.

7 Réseau de transport noyau

Le réseau de transport noyau (KTN) transporte les invocations et les réponses entre les nœuds DPE. Le présent paragraphe fournit un certain nombre de directives concernant la construction d'un réseau KTN, mais n'impose aucune limitation en ce qui concerne le mappage du réseau KTN vers la technologie de réseau sous-jacente.

7.1 Couche de messagerie

Le transport d'invocations entre nœuds DPE se fait au moyen d'un protocole de messagerie normalisé. Le protocole général entre courtiers ORB (GIOP) tel qu'il est défini dans l'architecture CORBA 2 [6] est spécifié comme protocole de messagerie obligatoire afin de garantir l'interfonctionnement entre nœuds DPE. L'utilisation du protocole GIOP est toutefois impossible ou déraisonnable dans certains environnements. L'architecture CORBA 2 décrit, dans ce cas, une démarche de spécification d'un protocole entre courtiers ORB propre à l'environnement (ESIOP). L'architecture DPE adopte une démarche similaire.

La spécification du protocole GIOP se constitue des éléments suivants:

- définition d'une représentation de données commune (CDR, *common data representation*): la représentation CDR constitue un mappage de transfert de syntaxe de types de données IDL OMG vers une représentation isomorphe à bas niveau; ce mappage est utilisé pour le transfert "sur la ligne" entre des courtiers ORB et des passerelles (agents) entre courtiers ORB;
- formats des messages GIOP: les messages GIOP sont échangés entre agents pour faciliter les demandes d'objet, localiser les implémentations de ces derniers et gérer les canaux de communication;
- hypothèses de transport GIOP: la spécification du protocole GIOP décrit des hypothèses générales s'appliquant à toute couche de transport réseau utilisable pour le transfert de messages GIOP. La spécification décrit également la manière dont sont gérées les connexions et les contraintes d'ordre de succession des messages GIOP.

7.2 Couche Transport

Un certain nombre de protocoles de transport sont utilisables pour le transfert des messages GIOP. On a spécifié un mappage du protocole GIOP vers un protocole de transport obligatoire afin de garantir l'interfonctionnement entre nœuds DPE. Ce protocole obligatoire est le protocole TCP/IP et le mappage du protocole GIOP vers ce dernier est appelé "protocole Internet entre courtiers ORB" (IIOP). La spécification du protocole IIOP de l'architecture CORBA 2 [6] est adoptée pour l'architecture DPE.

La spécification du protocole IIOP ajoute l'élément suivant à la spécification du protocole GIOP:

- transport de message avec protocole IOP Internet – La spécification du protocole IIOP décrit la manière dont les agents ouvrent des connexions TCP/IP et les utilisent pour le transfert de messages GIOP.

Le protocole IIOP ne correspond pas à une spécification proprement dite, mais constitue une spécialisation ou un mappage du protocole GIOP vers le protocole TCP/IP de transport spécifique. La spécification du protocole GIOP (à l'exclusion de l'élément spécifique de transport IIOP) peut être considérée comme un point de conformité supplémentaire pour les mappages vers d'autres protocoles de transport.

7.3 Prescriptions pour l'environnement DPE

Il est clair que l'environnement DPE doit pouvoir prendre en charge des protocoles autres que le protocole IIOP pour le réseau KTN. Deux des piles de protocoles les plus usuelles devant être prises en charge pour les télécommunications sont la pile de protocoles SS7 (c'est-à-dire les couches des protocoles TCAP ou SCCP) et les protocoles de couche d'adaptation ATM 5 (AAL5). Il est nécessaire de spécifier des mappages du protocole GIOP vers ces protocoles pour la prise en charge de ces protocoles et garantir l'interfonctionnement.

NOTE – Les mappages du protocole GIOP vers d'autres protocoles appellent une étude ultérieure.

La question de l'interfonctionnement avec des protocoles multiples soulève le problème plus général de la prise en charge de plusieurs protocoles dans l'environnement DPE; cet environnement doit pouvoir communiquer en utilisant plusieurs protocoles dont les suivants:

- protocole GIOP;
- protocoles ESIOP;
- basés sur d'autres moyens de communication tels que des files d'attente de messages.

Les mécanismes de prise en charge de plusieurs protocoles doivent être transparents pour le réalisateur d'application. Ce dernier doit, toutefois, avoir la possibilité de positionner les paramètres liés à une connexion sur le réseau KTN. Le réalisateur d'application doit, par exemple, être en mesure de spécifier des prescriptions de largeur de bande pour une connexion de réseau KTN si le réseau sous-jacent prend en charge ce paramètre.

8 Interfonctionnement DPE

Le présent paragraphe fournit le cadre d'interfonctionnement général pour l'environnement DPE. Il décrit les prescriptions d'interfonctionnement à divers niveaux ainsi que le contexte pour le réseau de transport noyau (KTN).

La Rec. UIT-T X.931 [5] définit les détails du protocole de prise en charge des interactions informatiques. Elle traite en particulier des points suivants:

- définition d'un cadre d'interfonctionnement général (GIF, *general interworking framework*);
- définition, au sein du cadre GIF, d'une famille de primitives de service apparentées de manière fonctionnelle;
- spécification du mappage des primitives de service GIF et de leurs paramètres vers les messages et les champs de protocoles particuliers, y compris le protocole GIOP de l'architecture CORBA.

L'interfonctionnement DPE spécifie une démarche facilitant la prise en charge d'opérations entre des objets actifs sur des nœuds DPE différents. Cette démarche fournit une souplesse dans la mesure où elle permet des combinaisons pour la prise en charge de besoins propres à des environnements divers. Elle permet de respecter le concept de base d'un environnement homogène pour l'exécution des objets (services et applications).

Ce paragraphe traite de l'interfonctionnement au niveau du noyau DPE et des interactions entre ce niveau et le niveau communication.

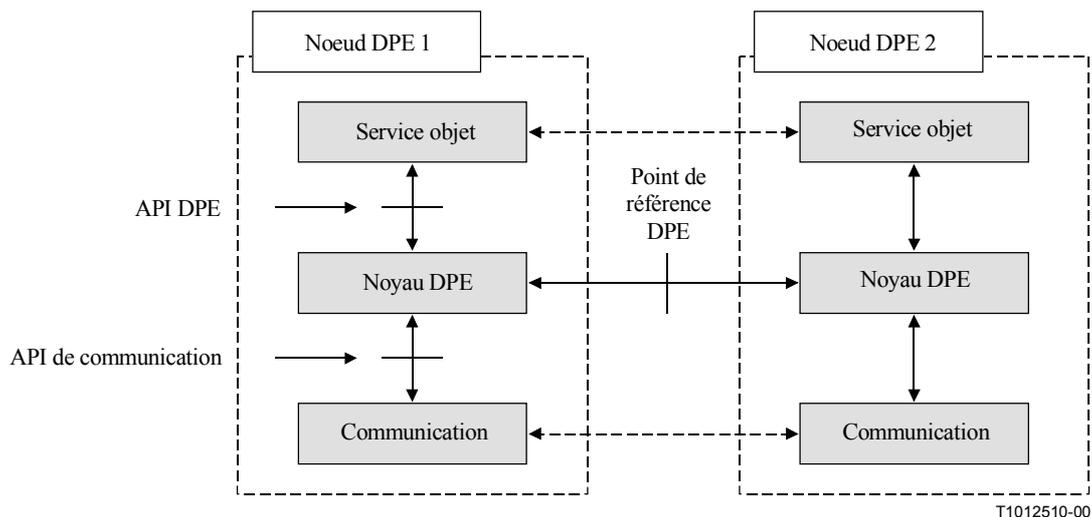


Figure 8/Z.600 – Cadre d'interfonctionnement général

8.1 Cadre d'interfonctionnement général

On peut distinguer les trois niveaux suivants pour l'interfonctionnement entre deux nœuds :

services objet : aucune information n'est fournie sur la manière dont les objets homologues perçoivent et traitent les informations lorsque les interfaces de service sont spécifiées au moyen du langage IDL ou ODL. La sémantique des services de base (c'est-à-dire, les objets "service") est en général bien connue, mais la sémantique interne des applications spécialisées est connue uniquement du concepteur de l'application. Le problème de l'interfonctionnement à ce niveau se pose en termes de sémantique interne des services ou des applications. Ce niveau d'interfonctionnement est en dehors du domaine d'application de l'environnement DPE;

noyau DPE : le transport des demandes de service et des réponses entre objets homologues se fait à ce niveau au moyen de mécanismes fournis par l'environnement DPE, impliquant en particulier :

- la liaison implicite ou explicite entre les interfaces des objets; cette liaison est prise en charge par des fonctions fournies par l'interface API de l'environnement DPE et par le point de référence de l'environnement DPE;
- le transfert de données impliquant la conversion des paramètres et des résultats des opérations vers un format de transmission commun;

communication : ce niveau est responsable de l'interfonctionnement entre les protocoles de transport utilisés pour la transmission des demandes de service et de leurs réponses au niveau de l'environnement DPE. Le niveau communication est en dehors du domaine d'application de l'environnement DPE.

La Figure 8 représente le cadre d'interfonctionnement général.

8.2 Extensions DPE de l'interfonctionnement CORBA 2

L'environnement DPE doit faire une distinction claire entre ses parties qui dépendent ou non du réseau. Cette séparation permet la réalisation d'une adaptation de protocole réseau indépendamment du noyau DPE. Une telle adaptation découple du reste de l'environnement la réalisation de protocoles réseau pour la communication entre les environnements DPE. Le mécanisme d'objets "protocole" enfichables permet l'intégration entre le noyau DPE et des modules d'adaptation réseau en provenance de fournisseurs divers. Il doit être possible de disposer de plusieurs adaptations réseau actives simultanément.

Les prescriptions suivantes étendent les possibilités d'interaction de l'architecture CORBA 2. Les implémentations de l'environnement DPE doivent prendre en charge les fonctionnalités suivantes pour une communication entre environnements DPE:

- sélection d'objets "protocole" pendant la configuration initiale d'un nœud DPE;
- sélection dynamique d'objets "protocole" en cours de fonctionnement d'un nœud DPE;
- insertion dynamique d'objets "protocole" (enfichables) en cours de fonctionnement d'un nœud DPE;
- sélection ou insertion dynamique d'objets "protocole" (enfichables) en fonction des besoins de qualité de service;
- possibilité d'utilisation simultanée d'objets "protocole" multiples.

Une interface de communication ouverte (OCI, *open communication interface*) a été définie en vue de compléter la définition de la communication entre courtiers ORB donnée par l'architecture CORBA 2 et fournir une base solide d'implémentation du réseau de transport noyau. La Figure 9 représente les relations entre les protocoles entre environnements DPE. Elle représente de manière plus précise l'utilisation du protocole IIOP sous la forme d'un protocole de transport enfichable.

Une démarche différente peut être utilisée pour les protocoles de messagerie dans le domaine des télécommunications. Dans un tel cas, le protocole de messagerie remplace effectivement le protocole GIOP de la couche Message de l'architecture CORBA. L'utilisation de la démarche ESIOP de l'architecture CORBA ESIOP permet également de définir un mappage pour des environnements utilisant le protocole TCAP (se référer à la Figure 9).

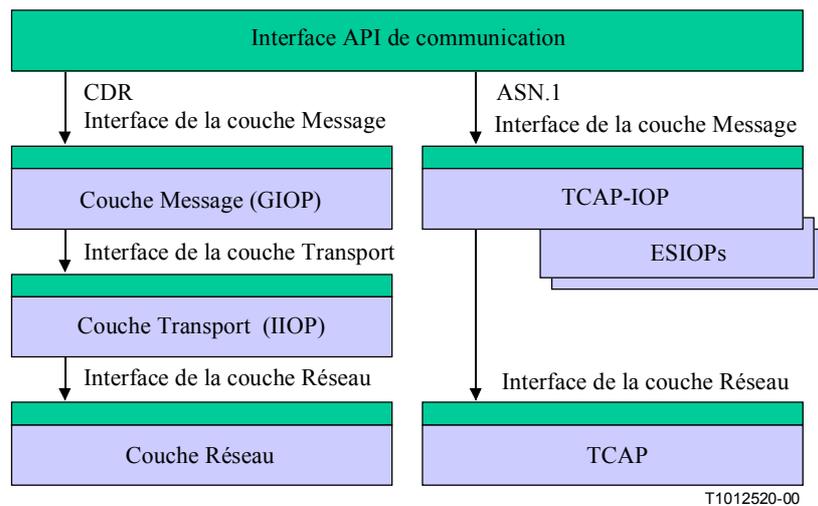


Figure 9/Z.600 – Exemple de mappages de protocole entre environnements DPE

9 Services d'ingénierie fournis par le noyau DPE

Ce paragraphe définit les services noyau DPE nécessaires à la prise en charge d'applications de télécommunications. Toutes les applications n'ont pas besoin de la totalité des services noyau DPE figurant dans la présente Recommandation. Il convient donc de définir des profils DPE prenant en charge diverses catégories de services et d'applications, chaque profil spécifiant un ensemble de services noyau obligatoires. La définition de tels profils appelle une étude ultérieure.

9.1 Introduction

Les prescriptions DPE sont classées selon les catégories suivantes:

- prescriptions de modularité;
- prescriptions fonctionnelles;
- prescriptions non fonctionnelles.

Les prescriptions de base devant être prises en charge par toutes les implémentations DPE sont associées au traitement d'invocations d'opérations des côtés d'objet homologues. L'architecture CORBA 2 satisfait à ces prescriptions et contient de son côté des prescriptions concernant l'interface d'invocation dynamique, l'interface d'invocation statique et l'adaptateur d'objet portable [6].

Un environnement DPE pour des systèmes de télécommunication doit prendre en charge une grande variété de systèmes:

- équipements réseau de niveau inférieur tels que des routeurs, des brasseurs, etc.;
- nœud de traitement de niveau supérieur;
- systèmes en temps réel;
- systèmes à hautes performances;
- systèmes à disponibilité élevée et avec tolérance de fautes.

Ce paragraphe présente les prescriptions et ne fait aucune hypothèse concernant le mode d'implémentation de l'environnement DPE au-dessus du système d'exploitation et les ressources matérielles sous-jacentes. Les possibilités d'implémentation vont de l'utilisation de bibliothèques et de serveurs normalisés au niveau application jusqu'à l'utilisation d'un environnement DPE totalement intégré au système d'exploitation qui le prend en charge. Les prescriptions données ici pour le noyau DPE vont, sous cet aspect, au-delà du domaine d'application d'un environnement DPE et impliquent des prescriptions concernant le système d'exploitation sous-jacent.

Un environnement DPE peut incorporer, en plus des prescriptions de base, tout service tel que le traitement des transactions ou de la sécurité.

NOTE – Certains services objet peuvent avoir un impact direct au niveau du noyau DPE. Ils ne peuvent pas être implantés sous la forme de services normalisés au-dessus de ce noyau, ou uniquement au prix d'une dégradation inacceptable des performances. La prise en charge de tels services appartient au domaine d'application des présentes prescriptions.

9.2 Caractéristiques concernant le temps réel

La présente Recommandation définit les caractéristiques suivantes pour un système en temps réel:

- spécification de contraintes temporelles, par exemple de délai et de priorité;
- spécification de prescriptions de capacité de traitement;
- garantie du respect des prescriptions précédentes;
- allocation et commande des ressources nécessaires pour assurer un comportement prévisible.

La nature des garanties fournies peut prendre l'une des formes suivantes:

- meilleur effort pour lequel le système ne fournit pas de garantie quantitative du niveau ou de la fréquence de respect des prescriptions de qualité de service;
- comportement déterministe pour lequel le système garantit que les prescriptions seront respectées de manière stricte pendant la durée de vie de l'application.

Toute garantie raisonnable fournie par un système peut être respectée uniquement pour des niveaux donnés de fiabilité de la qualité de service.

Il convient de faire une distinction importante entre des systèmes en temps réel ouverts ou fermés. Un système en temps réel fermé prend en charge un ensemble fixe d'applications connues à l'avance. Les systèmes en temps réel intégrés constituent un exemple. Un système en temps réel ouvert prend en charge des ensembles d'applications qui ne sont pas connues à l'avance et qui peuvent varier au cours du temps. Cette distinction est importante en ce qui concerne la nature des outils utilisés pour garantir la qualité de service fournie pour les applications. Les concepteurs d'une application de système en temps réel fermé peuvent planifier en temps différé l'ordonnancement et l'utilisation des ressources concernées. Cela n'est en général pas possible pour un système en temps réel ouvert pour lequel des garanties fortes (c'est-à-dire plus contraignantes que le meilleur effort) peuvent uniquement être fournies par le biais d'une certaine forme de contrôle d'admission en cours d'exécution.

Le fourniture de garanties de temps et de capacité de traitement au sein d'un système réparti ouvert en temps réel est une tâche ardue. Il s'ensuit qu'un environnement DPE en temps réel doit appliquer un principe général de séparation entre les politiques et les mécanismes d'exécution et ne pas imposer de politique particulière de gestion de ressources. Il doit, au contraire, fournir une souplesse suffisante pour tolérer des politiques diverses et offrir des interfaces de gestion directe des ressources système telles que les processeurs, la mémoire et les communications.

9.3 Prescriptions concernant le modèle d'exécution

Lorsqu'un noyau DPE reçoit une demande, la politique de commutation entre les fils d'exécution dans un environnement avec fils d'exécution multiples ne doit pas perturber le comportement en temps réel des autres parties de l'application active sur un système d'exploitation.

Le concept d'exécution simultanée correspond au partage d'un processeur unique entre un certain nombre de tâches (ou "fils d'exécution") d'une manière permettant de minimiser le temps nécessaire à la commutation entre les fils d'exécution. Ceci se fait en partageant d'une manière aussi complète que possible l'environnement d'exécution du programme entre les divers fils d'exécution, afin de minimiser le volume des informations d'état stockées et restaurées lors de la commutation entre les fils d'exécution.

Lorsqu'un environnement DPE est mis en place sur une plate-forme physique, le modèle d'exécution doit permettre la personnalisation du comportement interne conformément aux prescriptions en temps réel:

- gestion des fils d'exécution sur le serveur (nombre de fils d'exécution, priorité, classe d'ordonnancement, taille de la pile d'exécution);
- gestion du modèle d'exécution (politique utilisée lors de l'arrivée d'une demande).

9.4 Prescriptions de modularité

9.4.1 Souplesse de l'architecture DPE

L'architecture de l'environnement DPE permet d'effectuer les actions suivantes:

- introduction de nouveaux mécanismes de liaison et de communication;
- ajout incrémental de services de transparence (sécurité, transaction, persistance, migration, etc.).

L'environnement DPE peut en outre assurer une charge à la demande pour les services noyau. Les interfaces API correspondantes de l'environnement DPE doivent être publiées à l'intention des réalisateurs d'application ou de système afin de leur permettre de dimensionner la plate-forme en fonction des besoins particuliers de leurs applications.

La modularité de l'architecture DPE apporte les avantages suivants:

- il est plus facile de satisfaire à des contraintes strictes de performance et de taille s'il est possible de supprimer des composants modulaires plutôt qu'en procédant par dépouillement d'un environnement DPE monolithique;
- les coûts de réalisation (souvent pour un nombre réduit d'exemplaires) d'un environnement DPE destiné à une niche de marché ou à des marchés émergents sont réduits par la réutilisation de l'architecture et de bibliothèques constituées de composants remplaçables;
- l'interfonctionnement entre nœuds DPE est assuré parce que ces derniers sont des spécialisations d'une architecture DPE unique.

L'architecture DPE doit fournir les fonctionnalités suivantes:

- mécanismes d'abstraction généraux autorisant diverses politiques de gestion de ressources et de schéma de traitement, y compris celles qui permettent une commande au niveau application;
- mécanismes permettant aux concepteurs de combiner ces ressources. La conception de ces mécanismes d'abstraction doit être suffisamment modulaire pour fournir aux concepteurs le libre choix des composants lors de la construction d'un environnement DPE particulier. La modularité est associée ici aux propriétés classiques de simplicité, d'orthogonalité et de resserrement suffisant des interfaces.

9.4.2 Prise en charge de protocoles multiples

L'environnement DPE doit fournir les fonctionnalités suivantes:

- introduction et exécution simultanée de piles de protocole multiples convenant à divers types d'applications et divers besoins de qualité de service;
- capacités de choix dynamique de la pile de protocole utilisée et de commande du niveau de multiplexage des ressources de communication (par exemple, pour demander la création d'une liaison utilisant un canal de communication alloué précédemment);
- chargement à la demande du code du protocole.

Le § 9.4.1 pose le préalable implicite de la disponibilité de l'interface API de l'environnement DPE pour satisfaire à cette prescription. Les souches générées par le compilateur du langage IDL ou ODL doivent être indépendantes du protocole.

Il est possible, par exemple, de prendre en charge les protocoles suivants:

- protocoles de télécommunication: TCAP, CMIP/CMIS, SNMP, etc.;
- protocoles ORB et RPC: IIOP, DCE-CIOP, etc.;
- protocoles en temps réel pour le transport de contenus audio ou vidéo;
- protocoles de signalisation en temps réel.

La prise en charge de protocoles multiples constitue un besoin essentiel d'un environnement de télécommunication. Les anciens protocoles sont pris en charge ainsi que de nouveaux protocoles qui traitent de nouveaux besoins fonctionnels (par exemple, la diffusion multiple ou les communications de groupe).

9.4.3 Schéma de communication générique

Un environnement DPE incorpore un schéma de communication générique permettant la mise en œuvre de diverses politiques de multiplexage et la construction de tout protocole. Ce schéma fournit les fonctionnalités suivantes:

- combinaison des fonctions de protocole: il est possible de construire des piles de protocoles et de réutiliser des modules dans plusieurs piles. Il est nécessaire de minimiser les dépendances entre modules;

- la nature des piles de protocole est totalement découplée des problèmes de présentation de données. Il doit être possible, par exemple, d'utiliser une certaine politique de répartition avec n'importe lequel des protocoles de communication;
- configuration du protocole: les protocoles sont souvent propres à une plate-forme et doivent uniquement figurer dans une configuration lorsqu'une prise en charge nécessaire est disponible. La configuration des protocoles peut se faire au moment de la compilation ou de manière dynamique (lors de l'exécution);
- indépendance par rapport aux problèmes de simultanéité et de multiplexage, qui constituent la source la plus importante de charge fixe des protocoles et de rigidité de leurs architectures;
- schéma générique d'exécution simultanée permettant l'adaptation de divers procédés de gestion de la simultanéité;
- schéma générique de gestion de tampon permettant divers procédés de présentation de données, de fragmentation et de traitement de protocole.

Un schéma de communication générique facilite, pour les concepteurs et les réalisateurs de protocoles, l'extension de l'environnement DPE au moyen de nouveaux protocoles ainsi que la réutilisation d'implémentations normalisées de composants de protocoles. Ce schéma permet également un réglage fin des implémentations d'application normalisées en autorisant divers compromis de conception.

9.4.4 Prise en charge de liaisons souples

Un environnement DPE prend en charge tout type de liaison entre objets. Il doit en particulier:

- permettre la définition des liaisons par le réalisateur d'application;
- fournir des liaisons explicites (créées par le code d'application) et implicites;
- fournir la prise en charge de liaisons avec contraintes de qualité de service;
- prendre en charge tout scénario de liaison, incluant des liaisons multipoint pour une communication de groupe et des liaisons avec un tiers;
- permettre la gestion au niveau application pour des liaisons existantes.

L'Appendice II donne un exemple de liaison flexible.

9.5 Besoins fonctionnels

9.5.1 Prise en charge des interfaces de flux

L'environnement DPE doit fournir les fonctionnalités suivantes de prise en charge d'interfaces de flux:

- définition d'interfaces de flux avec un type strict;
- capacité de liaison entre plusieurs interfaces de flux;
- traitement des écoulements au niveau application;
- capacité de synchronisation entre plusieurs flux.

La notion d'interface de flux est nécessaire pour décrire des interfaces d'objets capables de générer ou de consommer des flux continus. L'introduction de ces interfaces nécessite la prise en charge des fonctionnalités suivantes:

- définition de signatures d'interface de flux;
- programmation en entrée et en sortie des flux d'informations associés aux interfaces de flux;
- commande des itinéraires de communication des flux d'informations entre diverses interfaces de flux;

- commande de la génération et de la réception des flux d'informations au niveau des interfaces de flux.

La commande des itinéraires de communication entre interfaces de flux, c'est-à-dire leur configuration, supervision et suppression, est prise en compte dans les prescriptions du § 9.4.4.

9.5.2 Prise en charge de l'exécution simultanée

Le noyau DPE doit fournir la prise en charge d'une exécution simultanée. L'environnement DPE doit fournir des fils d'exécution protégés et ré-entrants du côté client et du côté serveur. Les procédures de verrouillage utilisées pour la protection des structures de données de l'environnement DPE doivent être documentées et leur niveau de finesse doit être suffisamment détaillé pour atteindre un niveau de simultanéité donné. L'environnement DPE doit fournir une couche abstraite de fil d'exécution qui masque, pour le réalisateur, les procédures de fil d'exécution propres au système. Cette couche d'abstraction fournit les fonctionnalités suivantes:

- traitement des fils d'exécution (par exemple, leur création, leur destruction, etc.);
- ordonnancement de fils d'exécution: spécification de paramètres d'ordonnancement, y compris des paramètres liés au temps (par exemple des délais);
- fonctions de synchronisation: sémaphores, exclusion mutuelle, variables de condition, etc.;
- politique d'invocation et de communication: manière dont l'environnement DPE effectue la mise en attente des demandes et leur répartition vers le serveur;
- politique du serveur: manière dont une demande est traitée.

L'exécution simultanée est un besoin classique pour les systèmes répartis; elle est nécessaire pour obtenir les niveaux de simultanéité nécessaires aux applications de télécommunication réparties.

9.5.2.1 Ensemble de fils d'exécution

La définition correcte de la couche d'abstraction de fils d'exécution conduit à la définition d'un ensemble de fils d'exécution qui partagent des caractéristiques communes, parmi lesquelles:

- définition de paramètres de fils d'exécution:
 - paramètres POSIX communs (classe d'ordonnancement, taille de pile, priorité);
 - nombre maximal et minimal de fils d'exécution;
 - politique à appliquer lorsque tous les fils d'exécution ou tous leurs ensembles sont occupés (traitement d'exception ou mise en file d'attente de la demande).
- définition de la file d'attente des demandes concernant l'ensemble des fils d'exécution:
 - nombre maximal de demandes pouvant être mises en attente;
 - durée pendant laquelle une demande peut se trouver en attente de traitement. Une demande est extraite de la file d'attente et rejetée avec une exception lorsque cette durée d'attente est dépassée.

9.5.3 Mappage souple des événements vers les fils d'exécution

Les programmeurs d'application doivent pouvoir construire et définir des mappages entre les événements et les fils d'exécution. Les événements peuvent être des demandes, des signaux, etc. Les événements liés à des temporisations ou à un traitement de demande (par exemple, la réception d'une invocation) ont une importance particulière. L'environnement DPE doit permettre de spécifier des objets correspondant au mappage de demandes vers des fils d'exécution:

- un seul fil d'exécution par objet;
- une réserve dédiée de fils d'exécution, etc.

Il doit également permettre au réalisateur de gérer la distribution des demandes. Une politique usuelle de mappage de demandes en arrivée vers des fils d'exécution peut utiliser l'un des procédés suivants:

- la demande en arrivée sera traitée par le fil d'exécution actuel (en général par un protocole);
- la demande en arrivée sera traitée par un fil d'exécution nouvellement créé;
- la demande en arrivée sera traitée par un fil d'exécution choisi dans la réserve de fils d'exécution.

On peut concevoir un grand nombre de schémas divers de traitement des événements et d'implémentation des objets. Il n'existe pas une solution unique pour implémenter, avec une performance et une souplesse optimale, des objets communicants impliqués éventuellement dans plusieurs fils d'exécution. Il est nécessaire de tenir compte de divers compromis selon les fonctionnalités du système d'exploitation et les besoins de l'application. Il s'ensuit qu'un environnement DPE en temps réel doit permettre diverses implémentations et techniques de traitement. Cette fonction dépend en fin de compte de la possibilité d'associer de diverses manières des occurrences d'événements et des fils d'exécution responsables du traitement de ces occurrences.

9.5.4 Gestion de la simultanéité

Un serveur au sein d'une architecture client-serveur est en général en relation avec plusieurs clients et peut recevoir de ce fait des demandes qui se chevauchent. Les demandes seront mises en file d'attente si le serveur ne peut traiter qu'une seule demande à la fois, ce qui fait que les délais subis par les clients peuvent devenir excessifs si les temps de traitement sont importants ou si les demandes sont trop nombreuses. Il est prévu qu'un serveur sera normalement en mesure de traiter (ou de se comporter comme s'il traitait) plusieurs demandes en parallèle. Le serveur doit en outre traiter ces demandes simultanées avec une sécurité qui garantit l'absence de conflit entre les accès de données effectués par les demandes parallèles. Il s'ensuit que des mécanismes de simultanéité sont nécessaires au sein d'une capsule. Ce besoin correspond à une notion de "simultanéité allégée" et s'applique à des situations dans lesquelles la totalité des composants impliqués dans les activités parallèles résident dans un même espace d'adresse. La présente Recommandation ne traite pas de la simultanéité entre capsules.

9.5.4.1 Aperçu général du modèle de simultanéité

La simultanéité allégée peut être prise en charge par l'utilisation de mécanismes de **fil d'exécution** et de **tâche**. On peut la définir comme suit:

- une unité logique de simultanéité, appelée **fil d'exécution**, se constitue d'un itinéraire d'exécution indépendant à travers une application, pouvant être actif en parallèle avec d'autres unités logiques de simultanéité;
- une unité physique de simultanéité, appelée **tâche**, se constitue de l'ensemble des ressources nécessaires à l'exécution d'une unité logique de simultanéité.

Un fil d'exécution peut uniquement s'exécuter s'il est associé à une tâche. Les tâches sont partagées et réutilisées par les fils d'exécution au sein d'une application. Une fois que l'activité d'un fil d'exécution s'est achevée, la tâche sur laquelle il était actif est libérée et peut être utilisée par un autre fil d'exécution. La séparation des simultanéités logiques et physiques résultant de la définition de fils d'exécution et de tâches permet de gérer les ressources utilisées pour fournir la simultanéité dans une application sans affecter la simultanéité logique de l'application (se référer également au § 9.5.2).

Il peut y avoir à un instant donné plusieurs couples fil d'exécution/tâche liés se trouvant en attente d'exécution par le noyau DPE. Le noyau DPE doit organiser ces activités d'une manière ordonnée appelée communément "ordonnancement".

9.5.5 Processus générique d'ordonnancement

Un environnement DPE en temps réel doit offrir un cadre général d'ordonnancement qui sépare clairement d'une part les modules de politique d'ordonnancement pouvant être sélectionnés par l'application et, d'autre part, un ensemble partagé de mécanismes d'ordonnancement. Une prise en charge est nécessaire pour des politiques d'ordonnancement basées sur la priorité et des politiques d'ordonnancement basées sur le délai.

Les besoins en temps réel pour des applications dans un environnement ouvert peuvent être satisfaits avec un certain niveau de garantie statistique uniquement si l'ordonnancement des tâches qui implémentent ces applications peut utiliser des informations liées au temps, par exemple la durée d'exécution dans le cas le plus défavorable, les délais, la périodicité, etc. De telles informations sont absolument nécessaires pour le contrôle d'admission et peuvent être exploitées par des algorithmes d'ordonnancement (par exemple, délai le plus tôt en premier ou avec monotonie de débit). Comme il n'existe pas d'algorithme d'ordonnancement ou de politique de contrôle d'admission optimal, ni même de modèle de tâche en temps réel, il est nécessaire de sélectionner des politiques d'ordonnancement convenant aux applications.

9.5.6 Service de temps

Un service de temps dans un environnement DPE en temps réel doit fournir aux réalisateurs d'application le moyen d'obtenir l'heure actuelle avec une précision connue et d'utiliser des temporisations pour la gestion des événements en fonction du temps. Ce service est utilisé en général pour des applications avec contraintes de temps ainsi que pour activer la commande et la gestion en temps réel.

9.6 Besoins non fonctionnels

9.6.1 Degré de finesse des objets

L'environnement DPE doit pouvoir être mis à l'échelle afin de prendre en charge des applications prenant en charge un nombre important d'objets et de nombreuses connexions simultanées vers des objets distants.

Les coûts mémoire d'une référence d'interface distante non utilisée dans une capsule donnée doivent être comparables à ceux impliqués par un pointeur normal dans un langage de programmation.

La minimisation des coûts de communication distante lors du traitement de nombreuses connexions simultanées vers des objets dépend de la capacité du schéma générique de communication concernant le partage et la réutilisation des ressources de communication.

Un environnement de télécommunication comprend en général des objets qui ont des tailles très diverses en termes d'espace (taille mémoire) et de temps (durée de vie). Un environnement DPE doit prendre en charge des objets de tailles diverses et minimiser les coûts liés au traitement des objets de taille et de durée réduites, ainsi que ceux liés à la communication avec des objets distants.

9.6.2 Empreinte mémoire de taille réduite

La taille d'un environnement DPE en temps réel doit être suffisamment réduite afin que son empreinte mémoire permette sa mise en place sur des plates-formes de taille minimale, par exemple des systèmes intégrés des boîtes de dessus de poste ou des assistants numériques personnels. La taille mémoire impliquée par l'utilisation de l'environnement DPE en temps réel doit être documentée correctement et réduite à un minimum.

NOTE – Une détermination exacte de la taille mémoire mentionnée ci-dessus n'est pas nécessairement aisée, du fait des divers facteurs devant être pris en compte. On peut faire une première estimation en considérant les coûts mémoire associés à l'utilisation de l'environnement DPE en temps réel liés à la prise en charge d'une communication entre objets situés dans deux capsules différentes et en les comparant à ceux de la fonctionnalité native de communication fournie par le système d'exploitation sous-jacent.

9.6.3 Documentation du comportement temporel

L'environnement DPE doit fournir des informations concernant le comportement temporel de l'infrastructure de prise en charge afin de fournir aux applications des garanties de qualité de service. Les applications doivent connaître les latences caractéristiques des mécanismes de prise en charge pour des conditions de charge et d'environnement données. Ce besoin est comparable à la nécessité de connaître les primitives de base du système d'exploitation (pour une implémentation donnée).

Pour un environnement donné de matériel et de logiciel de base (c'est-à-dire un système d'exploitation), une implémentation DPE en temps réel doit être documentée au moyen d'une liste complète de mesures de performances et de contraintes de verrouillage pour toutes les fonctions DPE noyau (par exemple, les délais pour les itinéraires d'exécution dans les cas les plus défavorables, les durées des sections critiques, etc.). Il est nécessaire d'identifier et de documenter les latences caractéristiques d'environnement DPE en temps réel.

9.7 Interfaces multiples

L'environnement DPE doit prendre en charge des objets informatiques avec des interfaces multiples. Certaines de ces interfaces peuvent être créées en même temps que l'objet, alors que d'autres peuvent être créées de manière dynamique pendant la durée d'exécution de l'objet. Les interfaces multiples fournies par un objet peuvent être soit de types différents, soit des instances multiples du même type, soit une combinaison des deux.

Un objet peut utiliser différents types d'interface pour:

- fournir des services ayant des natures d'interaction différentes;
- fournir des services logiquement distincts;
- indiquer des droits d'accès différents pour divers utilisateurs.

Un objet peut utiliser des interfaces différentes du même type pour:

- fournir simultanément le même service à plusieurs clients avec une interface pour chaque client.

NOTE – Un objet informatique ne fournit pas nécessairement une interface différente pour chacun de ses clients. Il est possible, dans de nombreux cas, d'utiliser la même interface pour l'accès simultané de plusieurs clients.

La Rec. UIT-T Z.130 "Langage de définition d'objet de l'UIT" (ITU-ODL, *ITU object definition language*) [4] prend en charge la spécification d'interfaces multiples.

10 Services objet

Ce paragraphe décrit des services objet nécessaires pour la prise en charge de l'exécution d'applications de télécommunication. Certains de ces services objet sont d'une nature suffisamment générale pour être considérés comme des services informatiques d'utilisation générale, alors que d'autres sont propres au domaine des télécommunications.

10.1 Introduction

L'environnement DPE est une infrastructure qui fournit la prise en charge d'applications de télécommunication. Ceci nécessite que les services de l'environnement DPE fournissent des fonctions de prise en charge des mécanismes d'interaction entre objets. Ces services sont suffisamment génériques pour que leur description fasse partie de l'environnement DPE. Ils possèdent une interface informatique, c'est-à-dire qu'ils peuvent être invoqués comme s'ils étaient des objets informatiques.

Les objets "services" suivants sont actuellement considérés comme des objets "services DPE".

service de cycle de vie: le service de cycle de vie fournit des fonctions permettant de créer, de supprimer, de copier et de déplacer des objets individuels ou des ensembles d'objets. Il fournit également des capacités de désactivation, de réactivation, de duplication, de récupération et de migration;

service de dénomination: le service de dénomination est un service DPE fondamental. Il a pour objectif la localisation d'interfaces, en particulier pour des implémentations d'objet. Le service de dénomination fournit un mappage entre un nom lisible par un être humain et une référence d'interface;

service d'échange commercial: le service d'échange commercial prend en charge la liaison tardive entre les objets "exportateur" et "importateur". Il administre, pour ce faire, des informations concernant les offres de service ainsi que les références d'interface et les attributs de service correspondants. L'exportateur offre ses services (interfaces); l'importateur recherche des services et utilise le service d'échange commercial pour les détecter;

service de sécurité: le service de sécurité gère la confidentialité, l'intégrité, la responsabilité et la disponibilité au sein de l'environnement DPE. Le service de sécurité s'oppose aux menaces de divulgation, de tromperie, de perturbation ou d'usurpation portant sur des données et des services de télécommunication;

service de notification: le service de notification fournit à des objets la capacité d'émettre et de recevoir des notifications sans connaître l'ensemble des objets impliqués dans la communication. Il permet de même à des objets de recevoir des notifications sans interagir avec les objets émetteurs. Le service joue un rôle de courtier entre émetteurs et destinataires;

service de transaction: le service de transaction fournit les communications par transaction entre des objets, avec une garantie de la cohérence des applications utilisant certaines propriétés appelées dans leur ensemble propriétés ACID (atomicité, cohérence, isolation et durabilité). Des fonctionnalités de transactions ouvertes sont fournies pour des applications en temps réel;

service de gestion de la simultanéité: le concept de simultanéité est d'une importance considérable pour un système en temps réel, lorsqu'il est nécessaire de trouver un compromis entre une disponibilité élevée et la cohérence des applications. Le service de gestion de la simultanéité (CCS, *concurrency control service*) fournit la coordination entre les accès de fils d'exécution multiples à des objets qu'ils partagent. Le service effectue une conciliation lorsque plusieurs fils d'exécution accèdent simultanément à un objet partagé, de manière à préserver la cohérence de l'état de cet objet;

service de persistance: le service de persistance permet la gestion de la persistance de l'état des objets. Il assure l'intégrité des données d'une base de données en garantissant la cohérence entre les objets;

service d'interrogation: le service d'interrogation fournit des fonctionnalités d'interrogation portant sur des ensembles d'objets. Il peut être utilisé pour renvoyer des ensembles d'objets qui sont, soit choisis au sein d'un ensemble source pouvant être interrogé, soit produits par un processeur de demande. Les interrogations sont formulées dans un langage de requête et peuvent effectuer des opérations générales de traitement telles que la sélection, la mise à jour, l'insertion et la suppression dans des ensembles d'objets;

service de messagerie: les objectifs de haute performance, de mise à l'échelle et de capacité de traitement des applications de télécommunication doivent utiliser une communication asynchrone combinée à un modèle de programmation basé sur des événements. Le service de messagerie satisfait au besoin d'un modèle d'invocation avec une méthode véritablement asynchrone;

service de migration: un service de migration faisant partie du service de cycle de vie des objets de l'environnement DPE nécessite la réalisation de deux transparences vis-à-vis de la répartition:

- la transparence vis-à-vis de la relocalisation masque la migration d'un objet pour d'autres objets qui lui sont liés;

- la transparence vis-à-vis de la migration masque les changements d'emplacement pour l'objet concerné par la relocalisation.

service de licence: le service de licence permet la comptabilisation des accès et de l'utilisation pour des services de télécommunication et des applications logicielles;

journal d'événements: la gestion de tout système nécessite la capacité de pouvoir retracer l'historique des activités du système. Il existe de ce fait un besoin de base pour conserver de manière persistante les notifications d'événement et de pouvoir les stocker (dans un journal);

topologie: le service de topologie fournit un service général de gestion des relations topologiques (ou associations) entre des objets répartis. Il a pour but de décharger les objets application de la nécessité de gérer les associations, en fournissant un service de stockage d'informations de topologie indépendant d'une application spécifique;

distribution et installation de logiciel: le service d'installation et de distribution de logiciel fournit une prise en charge en temps réel de la distribution et/ou de l'installation du logiciel sur un nombre important de nœuds. L'environnement DPE est nécessaire, par exemple, pour la coordination correcte de la mise en place de nouveaux services de télécommunication;

gestion de configuration logicielle: le service de gestion de configuration logicielle permet de configurer divers services actifs dans l'environnement DPE, d'une manière comparable à l'utilisation de la fonctionnalité de gestion dans les réseaux de télécommunication pour le positionnement et la modification de paramètres des équipements physiques. L'ajout de telles fonctionnalités à l'environnement DPE permet aux utilisateurs de gérer les entités logicielles et matérielles d'une manière homogène. Ce service peut faire partie du service de gestion intégrée;

gestion intégrée: le service de gestion intégrée vise l'intégration de la gestion d'application utilisateur, de la gestion de plate-forme et de la gestion réseau;

commande et gestion de flux audio et vidéo: le document "Commande et gestion de services objet de flux audio et vidéo" [9] fournit une spécification détaillée de service.

NOTE – Les services objet peuvent être associés aux services et fonctions ODP de transparence vis-à-vis de la répartition (se référer à la Rec. UIT-T X.903 [3]). La réalisation effective de cette association appelle une étude ultérieure.

11 Bibliographie

- [7] OMG Document number formal/98-12-09, CORBA Services: Common Object Services Specification, Revised Edition, mars 31, 1995, Last Updated décembre 1998 (*Spécification des services objet communs*).
- [8] HOLZ (E.), KATH (O.), GEIPL (M.), LIN (G.), VOGEL (V.): The CAMOUFLAGE Project – Introduction of TINA into Telecommunications Legacy Systems, in *Proceedings of TINA'97 (Le projet CAMOUFLAGE – Introduction de l'architecture TINA dans des systèmes anciens de télécommunications)*.
- [9] OMG Document Number telecom/97-05-07, Control and Management of Audio/Video Streams, mai 30, 1997 (*Gestion et commande de flux audio et vidéo*).
- [10] GIFFORD (D.), GLASSER (M.): Remote pipes and procedures for efficient distributed communication, *ACM Transactions on Computer Systems*, Vol. 6, N° 3, 1998 (*Tuyaux distants et procédures pour une communication répartie efficace*).

- [11] COULSON (G.), BLAIR (G.S.), HORN (F.), HAZARD (L.), STEFANI (J.B.): Supporting the Real Time Requirements of Continuous Media in Open Distributed Processing, *Computer Networks and ISDN Systems, Special Issue on Open Distributed Processing*, Vol. 27, N° 8, juillet 1995 (*Prescriptions concernant un média continu dans le traitement réparti ouvert*).

ANNEXE A

Concepts de modélisation d'entreprise – Cadre général pour la propagation de prescriptions dans un marché de télécommunication ouvert

A.1 Domaine d'application

Les concepts de modélisation commerciale permettent de modéliser les services de télécommunication et d'informatique dans un environnement commercial avec compétiteurs multiples.

Les concepts de modélisation commerciale spécifient un cadre commercial général pour tous les compétiteurs au sein d'un marché de télécommunication ouvert.

L'instanciation du modèle commercial abstrait pour un service donné fournit les fonctionnalités suivantes:

- identification des rôles commerciaux nécessaires à la fourniture d'un service donné;
- association des rôles commerciaux avec les compétiteurs impliqués;
- identification des relations commerciales entre les rôles commerciaux et les domaines administratifs commerciaux détenus par les compétiteurs impliqués;
- spécification des points de référence qui implémentent les relations commerciales.

A.2 Termes et définitions

La présente annexe définit les termes suivants:

A.2.1 domaine administratif commercial: un domaine administratif commercial est défini par les besoins d'un ou de plusieurs rôles commerciaux et régi par un objectif commercial unique.

A.2.2 relation commerciale: association entre deux rôles commerciaux.

A.2.3 rôle commercial: fonction dont on estime qu'elle sera exécutée par un compétiteur dans un environnement commercial de télécommunication.

A.2.4 contrat: contexte qui définit des contraintes de fonctionnement pour un ou plusieurs points de référence.

A.2.5 point de référence: représentation d'une relation commerciale au sein du système de télécommunication. Le point de référence se constitue de spécifications liées à un point de vue et régies par un contrat.

A.2.6 compétiteur: participant qui a un intérêt ou une préoccupation commerciale dans l'activité commerciale de télécommunication. Un compétiteur est propriétaire d'un ou de plusieurs domaines administratifs commerciaux.

A.3 Concepts de modélisation commerciale

A.3.1 Cadre général

La base d'un système de télécommunication ouvert se constitue des objets informationnels, informatiques et d'ingénierie, qui appartiennent aux domaines administratifs commerciaux et sont séparés par des points de référence. La spécification des politiques et des interactions entre domaines administratifs commerciaux nécessite la spécification de la visibilité et les droits de chaque objet du domaine en considérant les domaines apparentés. La visibilité et les droits font partie d'un contrat. Le contrat est établi entre les domaines administratifs commerciaux et peut faire l'objet d'une négociation.

A.3.1.1 Contrat

Un contrat fournit la base pour les contextes définis dans les points de vue pris en charge. Les contextes des points de vue pris en charge peuvent être modifiés par une négociation dans le cadre des contraintes spécifiées dans le contrat. Le contrat ne peut toutefois pas être modifié à la suite des négociations effectuées par les points de vue pris en charge, car un point de vue donné fournit uniquement une vue partielle des interactions entre les domaines administratifs commerciaux et peut, de ce fait, violer des politiques négociées pour d'autres points de vue.

A.3.1.2 Domaine administratif commercial

Un domaine administratif commercial est défini par les besoins d'un ou de plusieurs rôles commerciaux. Les domaines administratifs commerciaux interagissent mutuellement par le biais de points de référence qui sont les implémentations des relations commerciales entre domaines administratifs commerciaux.

Le concept de domaine administratif commercial se base sur la notion de propriété. La propriété implique le privilège universel de gestion des entités au sein du domaine.

A.3.1.3 Rôles commerciaux et relations commerciales

Les rôles commerciaux sont mis en évidence par l'analyse des besoins commerciaux actuels et futurs des services d'information et de télécommunication. La définition des rôles commerciaux est illustrée par les types suivants de classification commerciale:

- **technique**: des domaines dans lesquels la technologie évolue à un rythme différent sont classés dans des rôles commerciaux différents;
- **économique**: les rôles commerciaux qui sont considérés comme des consommateurs ou des producteurs de services dans le marché actuel de l'information se voient attribuer des rôles commerciaux différents;
- **réglementaire**: certaines classifications des rôles commerciaux résultent de contraintes réglementaires.

Tout rôle commercial joue un rôle d'utilisateur ou de fournisseur vis-à-vis de certains autres rôles commerciaux. Le rôle de fournisseur ou d'utilisateur est déterminé par le contrat qui régit l'interaction entre les rôles commerciaux.

Les rôles commerciaux peuvent être combinés sous la forme de domaines administratifs commerciaux afin de satisfaire aux besoins du compétiteur pour son activité particulière.

Une relation commerciale exprime les besoins d'interaction entre deux rôles commerciaux. La relation commerciale entre deux domaines administratifs commerciaux se manifeste au niveau du point de référence.

A.3.1.4 Point de référence

Le point de référence se constitue de plusieurs spécifications liées à des points de vue et régies par un contrat. Une spécification de point de référence est subdivisée en segments de point de référence. Chaque segment correspond à une spécification significative et cohérente.

Le point de référence se constitue de l'ensemble des spécifications de tous les points de vue pris en charge. Il englobe les spécifications suivantes:

- **partie commerciale:** délimitation du domaine d'application, prescriptions fonctionnelles et non fonctionnelles imposées par les rôles commerciaux à la relation commerciale. Cette partie est déduite des interactions entre les rôles commerciaux;
- **partie informationnelle:** définition des informations partagées entre les domaines administratifs commerciaux;
- **partie informatique:** définition des interfaces des objets informatiques dont l'accès doit être fourni pour l'autre domaine;
- **partie d'ingénierie:** définition de la structuration de l'infrastructure de prise en charge répartie sous la forme de nœuds, de liaisons de signalisation et de commande, de systèmes d'exploitation assurant la prise en charge et de piles de protocole nécessaires aux interactions entre les domaines administratifs commerciaux;
- **partie diverse:** définition des autres contraintes, par exemple des limitations concernant d'autres spécifications importées par une spécification de point de référence, des limitations de conformité autorisées, etc.

A.3.2 Segmentation des points de référence

Une spécification de point de référence fait l'objet d'une segmentation. Un segment de point de référence est un sous-ensemble significatif et cohérent d'une spécification de point de référence.

La segmentation en accès et utilisation est motivée par la séparation entre les fonctionnalités de commande et de gestion des interactions du domaine administratif commercial (fonctionnalité d'accès) d'une part, et les fonctionnalités de fourniture et de gestion des services (fonctionnalité d'utilisation) d'autre part.

Un point d'accès se constitue des segments génériques suivants:

- segment d'accès;
- segment d'utilisation.

Le segment d'utilisation se constitue en outre des segments suivants:

- segment primaire;
- segment auxiliaire.

La segmentation du point de référence dépend de la combinaison de relations commerciales présente au niveau du point de référence et des fonctionnalités introduites effectivement par les relations commerciales dans le point de référence. Il peut exister d'autres segments.

A.3.2.1 Segments d'accès et d'utilisation

Voir Figure A.1.

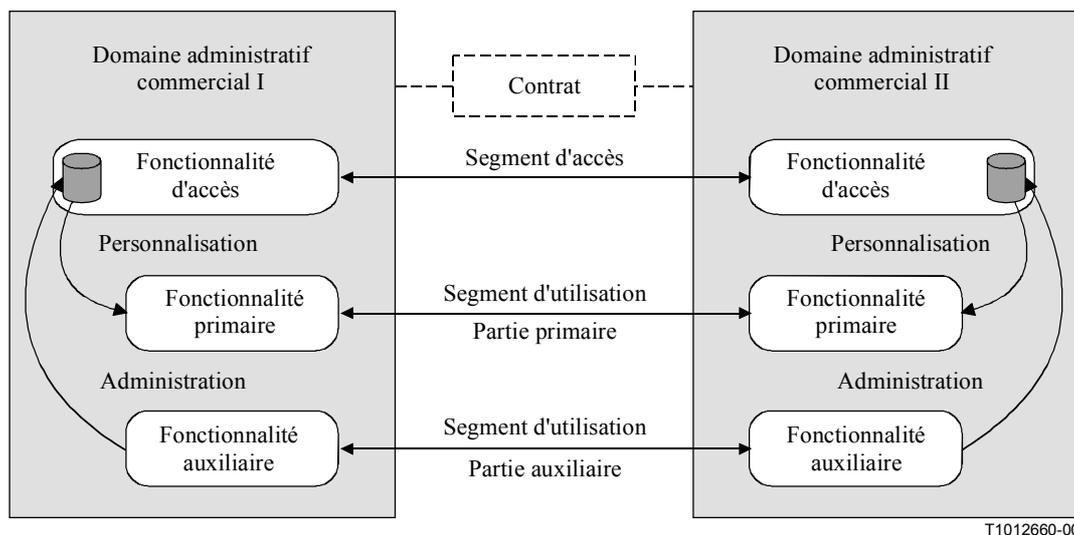


Figure A.1/Z.600 – Segmentation du point de référence générique

Le segment d'accès d'un point de référence fournit les fonctionnalités suivantes:

- initialisation du dialogue entre les domaines administratifs commerciaux;
- identification mutuelle des domaines administratifs commerciaux;

NOTE – Il convient de noter que l'un ou l'autre des domaines administratifs peut rester anonyme, selon l'interaction demandée.

- établissement d'une association sécurisée entre les domaines administratifs commerciaux;
- établissement du contexte de commande et de gestion de la fonctionnalité d'utilisation:
 - le contexte spécifie quels sont les services offerts et les conditions de l'offre;
 - le contexte peut être modifié de manière dynamique;
- initialisation du segment d'utilisation du point de référence entre les domaines administratifs commerciaux.

A.3.2.2 Segments primaires et auxiliaires

Le segment d'utilisation est propre aux services fournis effectivement entre les domaines administratifs commerciaux. Le segment d'utilisation peut faire l'objet d'une segmentation supplémentaire compte tenu des caractéristiques suivantes:

- utilisation (directe, par exemple, le service de vidéo à la demande (VoD, *Video-on-demand*) ou indirecte, par exemple, la gestion de fautes pour la vidéo à la demande);
- impact sur le domaine (par exemple, la fourniture, la gestion ou l'administration).

Le segment d'utilisation d'un point de référence est segmenté comme suit:

- fonctionnalité d'utilisation primaire;
- fonctionnalité d'utilisation auxiliaire.

Le segment d'utilisation primaire concerne l'objectif principal du contrat entre les domaines administratifs commerciaux. L'utilisation primaire du segment d'un point de référence fournit les fonctionnalités suivantes:

- gestion du cycle de vie du service;
- échange de contenu du service.

Le segment d'utilisation auxiliaire englobe la fonctionnalité de prise en charge des segments d'utilisation primaire ou d'accès. Les fonctionnalités suivantes sont fournies par le segment d'utilisation auxiliaire du point de référence:

- positionnement et gestion du contexte pour un service spécifique ou un ensemble de services spécifiques;
- positionnement et gestion du contexte pour l'administration du domaine, c'est-à-dire les fonctions qui ne sont pas propres à un service ou à un ensemble de services uniques;
- exécution des fonctions d'administration;
- exécution des fonctions de gestion;
- gestion du cycle de vie et des attributs d'utilisation des services auxiliaires.

Le segment d'utilisation auxiliaire ne participe pas à la réalisation du but contractuel primaire d'une relation commerciale; il ne possède pas de valeur indépendante, mais ajoute des valeurs au segment d'utilisation principale. Le segment d'utilisation auxiliaire peut modifier les informations et les politiques utilisées pour la prise de décisions dans le segment d'accès.

Diverses versions de relations commerciales peuvent exister. Chacune de ces versions différentes constitue un profil de la relation commerciale. Un profil est implémenté par le point de référence.

A.3.3 Combinaison de rôles commerciaux en domaines administratifs commerciaux

Comme indiqué par la Figure A.2, un ou plusieurs rôles commerciaux peuvent être combinés sous la forme d'un domaine administratif commercial unique appartenant à un seul compétiteur.

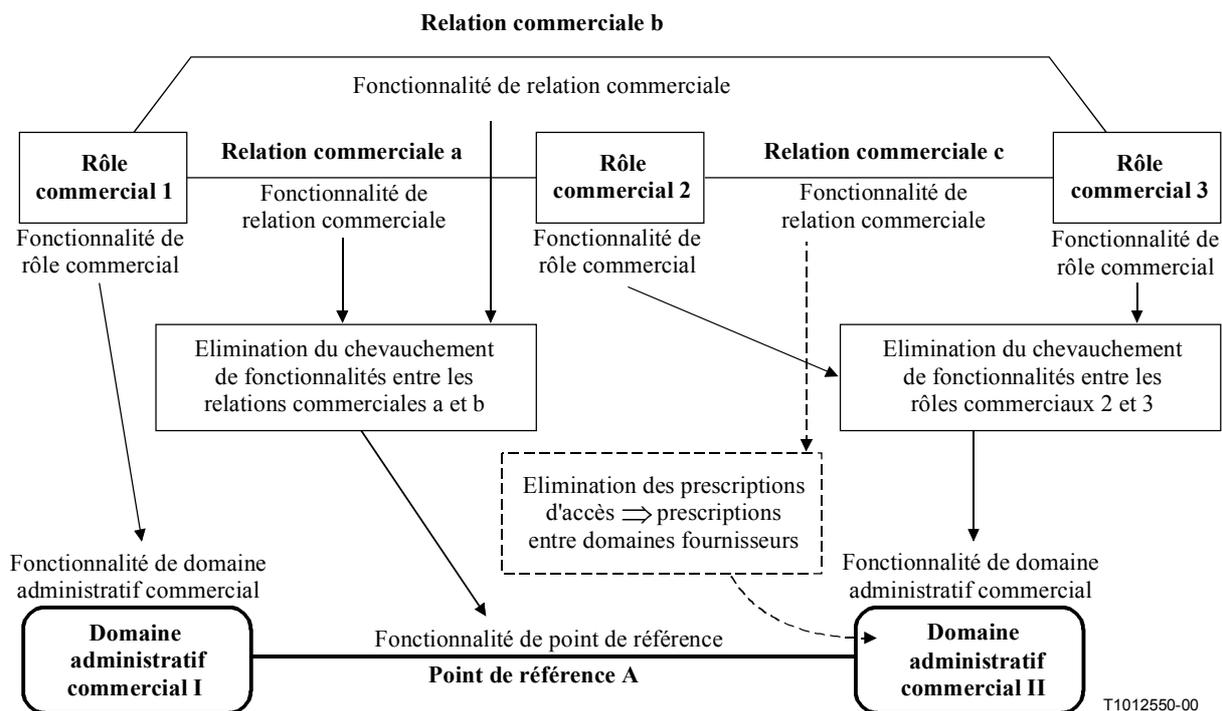


Figure A.2/Z.600 – Combinaison des rôles commerciaux en domaines administratifs commerciaux

L'interaction entre les rôles commerciaux 1, 2 et 3 est exprimée par les relations commerciales a (entre les rôles commerciaux 1 et 2), b (entre les rôles commerciaux 1 et 3) et c (entre les rôles commerciaux 2 et 3).

Le rôle commercial 1 s'effectue dans le domaine administratif commercial I; les rôles commerciaux 2 et 3 s'effectuent dans le domaine administratif commercial II.

Le point de référence A entre les domaines administratifs commerciaux I et II est défini comme correspondant à la combinaison des fonctionnalités des relations commerciales a et b, après l'élimination des recouvrements de fonctionnalités entre les relations commerciales a et b.

Le domaine administratif commercial II fournit de même les fonctionnalités combinées des rôles commerciaux 2 et 3 après l'élimination des recouvrements de fonctionnalités entre les rôles commerciaux 2 et 3.

L'implémentation de la relation commerciale c n'est pas visible à l'extérieur du domaine administratif commercial II. La relation commerciale c ne nécessite pas l'implémentation d'un point de référence. Cette implémentation peut se faire dans de nombreux cas au sein du domaine administratif commercial II en éliminant le segment d'accès, du fait qu'une commande d'accès n'est pas nécessaire au sein d'un domaine administratif commercial unique.

A.3.4 Délégation

La segmentation du point de référence fournit les fonctionnalités suivantes:

- réutilisation de segments de point de référence dans d'autres spécifications de point de référence;
- partage de fonctionnalités entre points de référence.

Ce procédé, illustré par la Figure A.3 entre les points de référence A et B, est appelé délégation.

La délégation peut être statique ou dynamique:

- la délégation statique est utilisée lorsque les segments délégués ne varient pas pendant la durée du contrat;
- la délégation dynamique est utilisée lorsque les segments délégués peuvent varier pendant la durée du contrat;

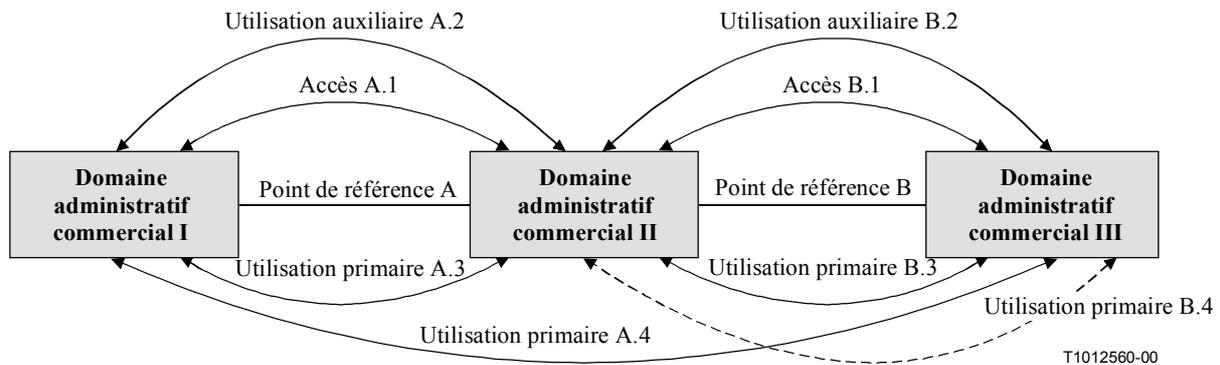


Figure A.3/Z.600 – Délégation de fonctionnalités de segment de point de référence

Le point de référence A entre les domaines administratifs commerciaux I et II est segmenté comme suit:

- accès A.1;
- utilisation auxiliaire A.2;
- utilisation primaire A.3;
- utilisation primaire A.4.

Le point de référence B entre les domaines administratifs commerciaux II et III est segmenté comme suit:

- accès B.1;
- utilisation auxiliaire B.2;
- utilisation primaire B.3;
- utilisation primaire B.4.

Le domaine administratif commercial II n'ajoute aucune valeur au segment de point de référence A.4. Le segment de point de référence B.4, qui fait partie de la relation commerciale entre les domaines administratifs commerciaux II et III, est fonctionnellement équivalent au segment A.4.

Le point de référence segment A.4 peut, si les contrats respectifs entre les domaines administratifs commerciaux I, II et III le permettent, être implémenté directement entre les domaines administratifs commerciaux I et III sous la forme d'un segment du point de référence B.

ANNEXE B

Concepts de modélisation d'ingénierie

Cette annexe fournit des explications concernant les relations entre les concepts de modélisation RM-ODP, dans la mesure où ces relations doivent être impliquées dans l'architecture DPE. Elle fournit également les références appropriées vers le modèle RM-ODP (Recommandations UIT-T de la série X.900) et le langage ODL de l'UIT (Rec. UIT-T Z.130).

NOTE – Le modèle RM-ODP spécifie un langage d'ingénierie, mais ne fournit pas de cadre général de mappage d'une spécification informatique, formulée en langage ODL ou IDL, vers le langage d'ingénierie. Il existe clairement une lacune méthodologique pour la prise en charge de ce mappage. Il est possible à l'heure actuelle de traduire directement les langages IDL et ODL vers un langage de programmation (par exemple, C++, Java, etc.) ([6] et [4]) en évitant de passer par le langage d'ingénierie. Ce mappage de langage direct introduit une dépendance de la spécification informatique à l'environnement de programmation cible, de sorte que la mise en place d'applications sur plusieurs nœuds dépend au minimum, des capacités de l'environnement cible. Le problème de mappage de la vue informatique vers la vue d'ingénierie appelle une étude ultérieure.

B.1 Concepts d'ingénierie de base

Le modèle RM-ODP définit une série de concept d'ingénierie qui joue un rôle fondamental. Il s'agit en particulier de l'utilisation d'objets, de capsules et de grappes au sein des nœuds; ces concepts sont traités ci-dessous et représentés dans la Figure B.1. Il est possible de mapper ces concepts vers des technologies d'objets répartis telles que l'architecture de courtier de demande d'objet commun (CORBA) ou d'autres technologies de communication d'objets propres au fournisseur.

Noeud DPE

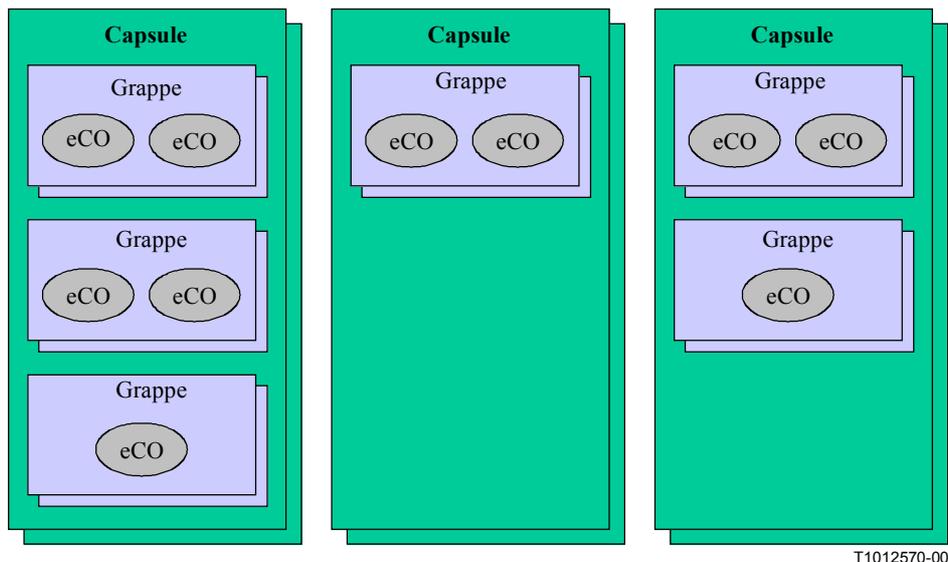


Figure B.1/Z.600 – Concepts d'ingénierie de base

Un objet informatique défini dans la Rec. UIT-T Z.130 se constitue d'un regroupement d'interfaces. Il est possible, durant la phase de conception d'un objet informatique possédant des interfaces multiples, d'en affiner la spécification et de le représenter sous la forme d'un ensemble d'objets informatiques dont chacun prend en charge une interface unique. La version 2.2 de l'architecture CORBA prend uniquement en charge la représentation d'objets informatiques avec une interface unique. La prise en charge d'objets informatiques avec des interfaces multiples peut se faire dans l'architecture CORBA par l'instanciation en temps réel d'objets auxiliaires d'ingénierie qui conservent la trace du regroupement des interfaces appartenant à un objet informatique avec interfaces multiples.

Noeud DPE: un noeud DPE est une représentation abstraite d'ingénierie d'un système informatique. Il modélise un ensemble de ressources qui effectuent des traitements autonomes. L'autonomie signifie que le noeud est en mesure de fonctionner indépendamment des autres noeuds; ce noeud DPE peut toutefois, dans la pratique, interagir avec d'autres noeuds et être géré par ces derniers. Le noeud DPE fournit le mécanisme de prise en charge de la transparence vis-à-vis de la répartition et masque, pour le réalisateur d'application, l'infrastructure informatique sous-jacente. Un noeud dispose des ressources informatiques suivantes:

- ressources de traitement telles que les processus, les tâches, les fils d'exécution ou le gestionnaire d'ordonnancement;
- ressources mémoire, par exemple en mémoire volatile ou en mémoire permanente;
- ressource de communication, par exemple les mécanismes de communication au sein d'un noeud (invocation IPC) et les mécanismes d'accès réseau (par exemple, modules de communication ou piles de protocole).

Un noeud DPE se constitue des parties suivantes:

- noyau DPE;
- services d'ingénierie;
- objets informatiques d'ingénierie (eCO);
- objets "protocole".

Le noyau DPE fournit un service de gestion de noeud accessible au niveau d'une ou de plusieurs interfaces de gestion de noeud.

Un nœud DPE peut interagir avec d'autres nœuds DPE au moyen d'un protocole normalisé. Le nœud DPE constitue une unité de connexion et de gestion réseau. Les responsabilités d'interconnexion et de gestion du réseau prennent fin au niveau de la frontière d'un nœud. Ce dernier reprend ces responsabilités pour ses connexions internes.

On peut donner les exemples suivants de nœud DPE:

- un ordinateur unique (avec unité centrale, mémoire et disque dur) exploité de manière indépendante;
- un système de traitement en parallèle;
- un système de traitement réparti utilisant un réseau local (LAN) et un système d'exploitation réparti;
- une grappe d'ordinateurs constituant un nœud à haute disponibilité.

Noyau DPE: un nœud DPE se trouve sous la commande du noyau DPE qui est responsable de l'initialisation du code, de la création des groupes d'objets, de la mise à disposition des ressources de communication et de la fourniture de services essentiels tels que la gestion du temps, la gestion de la simultanéité et l'ordonnancement. Le noyau DPE se constitue d'un ensemble de fonctions qui gèrent le nœud DPE. Le noyau DPE peut être implémenté au-dessus d'un environnement NCCE. Du point de vue de l'application ou de la gestion, seules les fonctionnalités abstraites fournies par le noyau DPE sont visibles à la place des fonctionnalités de l'environnement NCCE.

Dans un environnement DPE qui utilise, par exemple, diverses plates-formes propres aux fournisseurs, le noyau DPE correspondra à diverses spécialisations de ces types d'environnements NCCE.

Capsule: une capsule représente l'unité d'ingénierie pour l'allocation et l'encapsulation des ressources informatiques d'un nœud DPE. Un nœud contiendra en général plusieurs capsules. Une capsule peut être considérée comme une machine virtuelle ou un processus fournissant certaines garanties spécifiques de qualité de service (QS). Une capsule constitue une unité de gestion, de commande et de protection. Les objets mis en place dans cette unité partagent une même politique d'allocation des ressources fournies par le noyau d'un nœud DPE. Cette politique d'allocation est distincte de celle des autres capsules au sein du même nœud DPE.

Selon le mode de gestion des ressources du noyau, les ressources du nœud peuvent être partagées entre les capsules du nœud ou séparées pour une utilisation exclusive par chaque capsule. Dans ce dernier cas, les objets d'une capsule disposent de leur propre réserve de ressources et sont indépendants des objets situés dans les autres capsules. Dans le premier cas, les ressources sont partagées et les objets subissent les mêmes délais ou pénuries de ressources.

Un gestionnaire de capsule fournit le service de gestion de capsule au niveau d'une interface de gestion de capsule.

Une capsule contiendra en général plusieurs grappes.

Grappe: une grappe représente le plus petit regroupement d'objets possible. Le regroupement des objets d'une grappe permet de réduire leurs coûts de manipulation. Une grappe se constitue d'un ensemble d'objets eCO qui combine à un instant donné des propriétés d'emplacement, d'activation, de désactivation, de point de reprise, de réactivation, de récupération et de migration. Les grappes sont isolées dans une capsule. Un objet eCO peut appartenir à une grappe seulement.

La relation entre les objets eCO au sein d'une grappe est toujours locale. Les règles suivantes sont valables pour leurs interactions:

- **interactions entre grappes:** le concept de l'environnement DPE pour la modélisation des interactions d'objets eCO situés dans des grappes différentes est le canal;
- **interactions au sein d'une grappe:** les mécanismes d'interaction des objets eCO au sein d'une grappe ne font pas l'objet d'une prescription de l'environnement DPE.

Une unité d'instanciation ne constitue pas une propriété de la grappe; il est possible d'ajouter et de supprimer de manière dynamique des objets au sein d'une grappe.

Un gestionnaire de grappe fournit le service de gestion de grappe au niveau d'une interface de gestion de grappe.

Canal: un canal est un ensemble de mécanismes de prise en charge permettant la communication entre les grappes. L'ensemble de mécanismes nécessaires à la gestion et à la commande des interactions entre grappes fait partie d'un canal; ce dernier se constitue d'un certain nombre d'objets d'ingénierie qui interagissent (plus spécialement de souche, liant et objet "protocole").

APPENDICE I

Scénario du cycle de vie d'un objet

Le présent appendice traite du cycle de vie d'objet présenté dans le § 6.1.

Le cycle de vie d'un objet est en général le suivant:

- 1) une application est réalisée sous la forme d'un ensemble d'objets informatiques (CO) comprenant les codes source (par exemple, en langage C++) et les descriptions de structure et d'interface de l'objet (en langage ODL);
- 2) les codes source et les descriptions d'interface de l'objet sont traduits en code exécutable;
- 3) **mise en place:** le code exécutable de l'objet est mis en place au niveau d'un ou de plusieurs nœuds sous la forme (d'un ensemble) de squelettes d'objet eCO. Le fait de disposer du squelette d'objet eCO et du code d'exécution permet au nœud de créer une instance d'objet eCO. Des squelettes de grappe peuvent également être mis en place afin de permettre l'instanciation de grappes;
- 4) **création:** des ensembles d'objets eCO sont créés au sein des grappes et initialisés;
- 5) **activation:** les grappes sont activées en vue d'une exécution en allouant des ressources aux objets et en activant leurs interfaces;
- 6) **désactivation:** les grappes sont désactivées;
- 7) **suppression:** les grappes sont supprimées;
- 8) **retrait:** les squelettes d'objet eCO sont retirés.

L'étape 1 est prise en charge par un environnement de création de service qui n'est pas décrit par l'architecture DPE. L'étape 2 est prise en charge par des compilateurs, par exemple pour le langage C++, et par un compilateur de langage ODL.

L'architecture DPE fournit le modèle et les fonctions de prise en charge des étapes 3 à 8. Le service de distribution et d'installation de logiciel concerne les étapes 3 et 8 et le service de cycle de vie, les étapes 4 à 7.

APPENDICE II

Liaison flexible

Le présent appendice traite des liaisons présentées dans les § 6.4 et 6.6.

Une liaison dans un système informatique indique d'une manière classique un ensemble d'actions ou de processus permettant d'associer ou d'interconnecter divers objets de ce système. Le processus de

liaison implique l'établissement d'un itinéraire d'accès entre deux objets, lequel implique à son tour la localisation de la cible de l'itinéraire d'accès, la vérification des droits d'accès et l'établissement des structures de données adéquates nécessaires à la prise en charge de l'itinéraire d'accès utilisé pour la communication entre les objets sur cet itinéraire. La notion de liaison couvre la liaison au niveau du système d'exploitation, effectuée généralement entre deux espaces d'accès différents, et la liaison au niveau du langage qui établit en général un mappage entre une variable et un objet cible qui s'effectue au sein d'un même espace d'adresse. Dans le cas des systèmes répartis, le processus de liaison combine des fonctionnalités au niveau du système d'exploitation et une liaison au niveau du langage et englobe en plus la mobilisation de ressources de communication pour la prise en charge des interactions distantes (c'est-à-dire, entre nœuds DPE différents) entre les objets.

Les systèmes avec invocation de procédure distante (RPC) constituent un exemple de combinaisons de liaisons au niveau du système d'exploitation et du langage; la liaison établit des souches d'objet typées et la prise en charge d'une connexion vers un serveur; un autre exemple est fourni par des systèmes avec objets persistants, dans lesquels la liaison établit un objet "cache mémoire" qui est lui-même lié à un emplacement sur disque.

La fourniture d'une intégration souple entre les liaisons au niveau système et langage constitue un besoin important des systèmes répartis ouverts. Il existe, même dans le cas classique d'une architecture client-serveur, une grande variété de sémantiques d'invocation d'opération et de mécanismes d'implémentation qui correspondent à des besoins d'application divers. Les serveurs peuvent, par exemple, être persistants ou dupliqués ou utiliser des caches gérés par diverses politiques permettant d'accroître les performances, etc. D'une manière plus générale, en particulier pour les communications multimédia par ordinateur, les communications entre objets peuvent prendre diverses formes incluant des interactions avec sources et destinations multiples et l'échange de données de flux continus. Ces applications nécessitent en général la prise en charge d'une combinaison de divers procédés de communication et de protocoles spécialisés. Il a été noté depuis longtemps que l'invocation RPC ne fournit pas une bonne prise en charge de l'échange d'un flux de données permanent [10]; ceci est encore plus évident dans le cas de l'échange de flux continus, tels que les flux audio ou vidéo [11]. Ces applications nécessitent également une gestion explicite de la configuration des objets et, en général, de l'établissement des itinéraires de communication entre les objets.

Comme il est très peu probable qu'un mécanisme universel unique soit en mesure de traiter tous ces cas, un système réparti ouvert doit fournir un cadre général permettant de prendre en charge et de combiner divers types de liaison avec diverses sémantiques de communication.

APPENDICE III

Problèmes de prise en charge d'interfaces de flux

Le présent appendice traite des interfaces de flux présentées dans les § 6.6.2 et 6.6.3.

La prise en charge des interfaces de flux pose un certain nombre de problèmes:

- est-il nécessaire de définir des interfaces de flux, et si oui lesquelles?
- de quoi se constitue une interface de flux? Flux unique, flux multiples unidirectionnels, flux bidirectionnels, etc.?
- comment gérer des groupes d'interfaces de flux?
- quels sont les mappages de flux qui seront fournis pour les interfaces de flux?

Il est utile de faire une comparaison entre la prise en charge de flux multimédia et de liaison présentée dans le présent appendice et d'autres démarches telles que les services MSS IMA¹. La spécification du service MSS IMA fournit la prise en charge de flux continu et permet, de la manière suivante, la commande explicite de la configuration de flux multimédia entre objets: la spécification MSS modélise des équipements multimédia sous la forme d'objets avec interfaces multiples appelés équipements virtuels. Un équipement virtuel possède une interface de serveur appelée interface de commande (voir Note) qui fournit à ses clients les moyens permettant l'observation des emplacements de flux de média et la commande du flux de données de média. Un équipement virtuel contient également des ports correspondant à ses mécanismes d'entrée et de sortie. Les ports sont identifiés par un index au sein d'un équipement mais ne fournissent aucune interface visible. Chaque port est associé à une interface de format qui fournit une abstraction d'informations de type de données (par exemple, le type de codage, le format de trame, etc.). Les ports peuvent être reliés par des connexions virtuelles. L'instanciation d'une connexion virtuelle à partir d'un atelier donné fournit comme résultat une interface de commande d'un type similaire à celui d'une interface de commande d'un équipement virtuel. L'interface de commande de connexion virtuelle offre une opération de connexion permettant de lier un port de sortie vers un ou plusieurs ports d'entrée (en fonction du type de diffusion simple ou multiple de la connexion virtuelle).

Les concepts TINA et ODP pour la liaison d'objets se superposent au concept de connexion virtuelle de la spécification MSS et la notion TINA et ODP d'interface de flux généralise la notion de port dans la spécification MSS. Cette dernière ne répond toutefois pas à la prescription d'interface de flux exprimée dans le présent appendice. La spécification MSS n'indique pas comment implémenter les ports et comment y accéder. La prescription actuelle définit par contre des interfaces de flux programmables: les programmeurs sont, de ce fait, en mesure de fournir des implémentations d'interfaces de flux d'une manière très voisine de la fourniture des interfaces d'opérations. Comme exigé par le modèle RM-ODP, les interfaces de flux peuvent être traitées de la même manière que les interfaces d'opérations. Les ports de la spécification MSS ne sont, par contre, pas vus comme des interfaces et font l'objet d'un traitement spécial (par exemple, il est uniquement possible d'y faire référence au moyen d'index entiers au sein d'un équipement virtuel donné).

La prescription de possibilité de programmation des interfaces de flux s'appuie sur plusieurs motifs. La disponibilité de plates-formes d'utilisation générale et à faible coût avec des capacités multimédia permet le traitement logiciel de flux continu. Les programmeurs réalisant des applications qui traitent des flux (par exemple en améliorant un navigateur Web par l'ajout de signaux audio et vidéo en direct) doivent être isolés des caractéristiques de format de média à bas niveau dépendantes du protocole et des détails propres au matériel.

NOTE – La spécification MSS utilise le terme "objet flux" pour désigner l'interface de commande. Nous nous en tenons à la terminologie ODP pour éviter des confusions et utiliserons les termes "interfaces de commande MSS" au lieu des termes "objets de flux MSS".

¹ L'IMA (Association Multimédia Interactive) a réorienté ses efforts pour s'occuper moins de la création des normes techniques d'interopérabilité et plus de questions de propriété intellectuelle et d'autres sujets liés au marché, et a fusionné avec la Software Publishers' Association (Association des éditeurs de logiciels).

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