



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.901**

(08/97)

SÉRIE X: RÉSEAUX POUR DONNÉES ET  
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Traitement réparti ouvert

---

**Technologies de l'information – Traitement  
réparti ouvert – Modèle de référence:  
aperçu général**

Recommandation UIT-T X.901

(Antérieurement Recommandation du CCITT)

---

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

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

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

# **NORME INTERNATIONALE 10746-1**

## **RECOMMANDATION UIT-T X.901**

### **TECHNOLOGIES DE L'INFORMATION – TRAITEMENT RÉPARTI OUVERT – MODÈLE DE RÉFÉRENCE: APERÇU GÉNÉRAL**

#### **Résumé**

La présente Recommandation | Norme internationale fait partie intégrante du modèle de référence du traitement réparti ouvert (ODP, *open distributed processing*). Elle contient un aperçu général du modèle de référence ODP, en précise les motivations, le domaine d'application et la justification, avec une explication des concepts clés ainsi qu'une présentation de l'architecture ODP. Elle explique la façon d'interpréter ce modèle de référence et la manière dont il peut être utilisé, en particulier par les rédacteurs de normes et par les architectes de systèmes ODP. Elle contient également une classification des domaines de normalisation en matière de systèmes répartis; cette classification s'appuie sur les points de référence de conformité identifiés dans la Rec. UIT-T X.903 | ISO/CEI 10746-3.

#### **Source**

La Recommandation X.901 de l'UIT-T a été approuvée le 9 août 1997. Un texte identique est publié comme Norme internationale ISO/CEI 10746-1.

## AVANT-PROPOS

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

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

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

Dans certains secteurs 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 1998

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

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives ..... 1
2.1	Recommandations   Normes internationales identiques ..... 1
2.2	Paires de Recommandations   Normes internationales équivalentes par leur contenu technique ..... 2
2.3	Normes internationales ..... 2
3	Définitions..... 2
3.1	Définitions de la présente Recommandation   Norme internationale ..... 2
3.2	Définitions données dans d'autres Recommandations   Normes internationales ..... 2
4	Abréviations ..... 6
5	Conventions ..... 7
6	Normalisation ODP ..... 7
6.1	Objectifs et motivations ..... 7
6.2	Réalisation ..... 9
6.2.1	Modélisation par objets ..... 9
6.2.2	Spécifications de points de vue ..... 9
6.2.3	Transparences à la répartition ..... 10
6.2.4	Conformité ..... 10
6.3	Normes..... 10
6.3.1	Le modèle de référence ..... 10
6.3.2	Normes spécifiques ..... 11
7	Fondements ..... 11
7.1	Concepts de modélisation de base ..... 12
7.1.1	Objets ..... 12
7.1.2	Interfaces et points d'interaction ..... 12
7.1.3	Comportement et état ..... 12
7.2	Concepts de spécification ..... 13
7.2.1	Composition et décomposition..... 13
7.2.2	Compatibilité de comportement..... 13
7.2.3	Type et classe..... 14
7.2.4	Gabarits..... 14
7.2.5	Rôles ..... 14
7.2.6	Classes de base et classes dérivées..... 14
7.3	Concepts de structuration..... 15
7.3.1	Groupes et domaines..... 15
7.3.2	Désignation ..... 15
7.3.3	Contrat ..... 15
7.3.4	Liaison et relation de liaison ..... 16
8	Architecture..... 16
8.1	Cadre architectural..... 16
8.1.1	Points de vue..... 17
8.1.2	Transparences à la répartition ..... 17
8.2	Langage d'entreprise ..... 18
8.3	Langage d'information ..... 20
8.4	Langage de traitement..... 21
8.4.1	Interfaces de traitement..... 23
8.4.2	Modèle de liaison..... 23
8.4.3	Typage et sous-typage dans les interfaces de traitement..... 25
8.4.4	Portabilité..... 26

8.5	Langage d'ingénierie .....	27
8.5.1	Grappes, capsules et nœuds .....	27
8.5.2	Canaux .....	28
8.5.3	Références d'interface .....	31
8.5.4	Liaison .....	31
8.5.5	Etablissement d'un canal .....	31
8.5.6	Interfaces de gestion .....	32
8.5.7	Intercepteurs .....	32
8.5.8	Points de conformité .....	34
8.6	Langage de technologie .....	34
8.7	Cohérence des points de vue .....	35
8.7.1	Cohérence du point de vue entreprise avec les autres points de vue .....	36
8.7.2	Correspondances entre les spécifications de traitement et d'ingénierie .....	37
8.8	Fonctions ODP .....	39
8.8.1	Fonctions de gestion .....	39
8.8.2	Fonctions de coordination .....	40
8.8.3	Fonctions de conteneur .....	41
8.8.4	Fonctions de sécurité .....	41
8.9	Transparences ODP à la répartition .....	42
8.9.1	Transparence d'accès .....	42
8.9.2	Transparence aux défaillances .....	42
8.9.3	Transparence à la position .....	42
8.9.4	Transparence à la migration .....	42
8.9.5	Transparence à la persistance .....	43
8.9.6	Transparence à la relocalisation .....	43
8.9.7	Transparence à la duplication .....	43
8.9.8	Transparence aux transactions .....	43
9	Evaluation de la conformité .....	43
9.1	Evaluation de la conformité et processus de développement .....	43
9.2	Evaluation de la conformité: relations pertinentes .....	44
9.3	Points de conformité et concepts apparentés .....	44
9.4	Spécifications de conformité ODP .....	45
9.4.1	Niveau d'abstraction .....	45
9.4.2	Usage de plusieurs points de référence .....	46
9.5	Conséquences des langages de points de vue sur la conformité .....	46
9.6	Activités d'évaluation de la conformité .....	47
10	Gestion des systèmes ODP .....	47
10.1	Domaines de gestion .....	47
10.2	Politique de gestion .....	48
10.3	Modélisation des structures de gestion .....	48
11	L'emploi des normes dans les systèmes ODP .....	48
11.1	Point de vue entreprise .....	48
11.1.1	Spécification d'entreprise .....	48
11.1.2	Application des normes .....	50
11.2	Point de vue information .....	50
11.2.1	Spécification d'information .....	50
11.2.2	Application des normes .....	50
11.3	Point de vue traitement .....	51
11.3.1	Spécification de traitement .....	51
11.3.2	Application des normes .....	52
11.4	Point de vue ingénierie .....	52
11.4.1	Spécification d'ingénierie .....	52
11.4.2	Application des normes .....	52
11.5	Point de vue technologie .....	53
11.5.1	Spécification de technologie .....	53
11.5.2	Application des normes .....	54

	<i>Page</i>
12 Exemples de spécifications ODP .....	54
12.1 Système de téléconférence multimédia .....	55
12.1.1 Introduction.....	55
12.1.2 Spécification d'entreprise .....	56
12.1.3 Spécification d'information .....	57
12.1.4 Spécification de traitement.....	58
12.1.5 Spécification d'ingénierie.....	61
12.1.6 Spécification de technologie .....	62
12.2 Liaison flux audiovisuelle multiparties .....	62
12.2.1 Description générale .....	63
12.2.2 Spécification d'entreprise .....	63
12.2.3 Spécification d'information .....	65
12.2.4 Spécification de traitement.....	67
12.2.5 Spécification d'ingénierie.....	69
12.2.6 Spécification de technologie .....	71
12.3 Exemple de gestion – Objet métrique .....	71
12.3.1 Spécification d'entreprise .....	72
12.3.2 Spécification d'information .....	73
12.3.3 Spécification de traitement.....	74
12.4 Exemple de base de données.....	75
12.4.1 Spécification d'entreprise .....	75
12.4.2 Spécification d'information .....	76
12.4.3 Spécification de traitement.....	76
Annexe A – Bibliographie.....	78

## Introduction

La croissance rapide des applications réparties a fait naître le besoin d'un cadre pour coordonner la normalisation du traitement réparti ouvert (ODP, *open distributed processing*). Le modèle de référence ODP fournit ce cadre. Il établit une architecture qui permet la prise en compte de la répartition, de l'interfonctionnement et de la portabilité.

Le modèle de référence pour le traitement réparti ouvert (RM-ODP, *reference model of open distributed processing*), Rec. UIT-T X.901 | ISO/CEI 10746-1 à Rec. UIT-T X.904 | ISO/CEI 10746-4, repose sur des concepts précis issus des développements récents dans le domaine des traitements répartis et s'appuie, dans la mesure du possible, sur l'utilisation des techniques de descriptions formelles pour la spécification de l'architecture.

Le modèle de référence ODP (ISO/CEI 10746) se compose:

- de la Rec. UIT-T X.901 | ISO/CEI 10746-1: **aperçu général**: elle contient un aperçu général du modèle de référence ODP, en précise les motivations, le domaine d'application et la justification, et propose une explication des concepts clés, ainsi qu'une présentation de l'architecture ODP. Elle explique la façon d'interpréter le modèle de référence ODP et la manière dont il peut être utilisé, en particulier, par les auteurs de normes et les architectes de systèmes ODP. Elle contient également une classification des domaines de normalisation en matière de systèmes répartis; cette classification s'appuie sur des points de référence de conformité identifiés dans la Rec. UIT-T X.903 | ISO/CEI 10746-3. Ces textes communs ne sont pas normatifs;
- de la Rec. UIT-T X.902 | ISO/CEI 10746-2: **fondements**: elle contient la définition des concepts et le cadre analytique à utiliser pour la description normalisée de systèmes de traitement répartis (arbitraires). Elle s'en tient à un niveau de détail suffisant pour étayer la Rec. UIT-T X.903 | ISO/CEI 10746-3 et établir les exigences de nouvelles techniques de spécification. Ces textes communs sont normatifs;
- de la Rec. UIT-T X.903 | ISO/CEI 10746-3: **architecture**: elle contient la spécification des caractéristiques d'un système réparti ouvert. Ce sont les contraintes auxquelles les normes ODP doivent se soumettre. Elle utilise les techniques descriptives de la Rec. UIT-T X.902 | ISO/CEI 10746-2. Ces textes communs sont normatifs;
- de la Rec. UIT-T X.904 | ISO/CEI 10746-4: **sémantique d'architecture**: elle contient une formalisation des concepts de modélisation ODP définis dans la Rec. UIT-T X.902 | ISO/CEI 10746-2, articles 8 et 9. La formalisation s'obtient en interprétant chaque concept à partir d'éléments des différentes techniques normalisées de descriptions formelles. Ces textes communs sont normatifs.

La présente Recommandation | Norme internationale comporte une annexe.

L'article 6 explique en quoi les systèmes répartis ouverts intéressent les affaires et comment le modèle de référence ODP et les normes qui lui sont associées permettront aux entreprises d'en recueillir les bénéfices. Cet article présente les "promesses" de l'ODP, qui sont des composants opérationnels dès insertion (*plug-and-play*) et des outils d'intégration de système pour systèmes répartis.

Les articles 7 à 10 précisent ce que sont le modèle de référence ODP et ses fonctions réparties. Ces articles démontrent que le modèle de référence ODP donne les moyens de concevoir des composants opérationnels dès insertion et des outils d'intégration de système pour systèmes répartis.

L'article 11 montre comment la spécification d'un système ODP peut faire référence aux normes ODP et à des spécifications issues d'autres groupes. Ces relations forment un facteur clé de la faculté d'intégration de techniques disparates qui caractérise l'ODP.

L'article 12 contient des exemples qui illustrent l'usage du modèle de référence ODP et des principes qui le sous-tendent pour la résolution des problèmes rencontrés dans les affaires.

## NORME INTERNATIONALE

## RECOMMANDATION UIT-T

## TECHNOLOGIES DE L'INFORMATION – TRAITEMENT RÉPARTI OUVERT – MODÈLE DE RÉFÉRENCE: APERÇU GÉNÉRAL

### 1 Domaine d'application

La présente Recommandation | Norme internationale:

- présente l'ODP et en donne les motivations;
- offre un aperçu général du modèle de référence pour le traitement réparti ouvert (RM-ODP, *reference model of open distributed processing*) dont elle explique les concepts clés;
- indique comment appliquer le modèle de référence ODP.

La présente Recommandation | Norme internationale embrasse aussi bien l'aperçu général que des explications détaillées. Elle peut être consultée de diverses manières au cours de la lecture des normes:

- a) si vous comptez ne lire que la présente Recommandation | Norme internationale, souhaitant acquérir une compréhension générale de l'importance que revêt l'ODP pour votre organisation, concentrez votre attention sur l'article 6;
- b) si vous avez l'intention d'étudier le modèle de référence ODP dans sa totalité, vous devriez lire aussi l'article 6 avant d'attaquer les Rec. UIT-T X.902 | ISO/CEI 10746-2 et Rec. UIT-T X.903 | ISO/CEI 10746-3;
- c) pendant le cours de votre lecture des Rec. UIT-T X.902 | ISO/CEI 10746-2 et Rec. UIT-T X.903 | ISO/CEI 10746-3, vous pourrez être amené à désirer consulter les articles 7 à 10, où se trouvent des explications qui détaillent les divers concepts définis dans ces textes communs;
- d) lorsque vous aurez achevé une première lecture des Rec. UIT-T X.902 | ISO/CEI 10746-2 et Rec. UIT-T X.903 | ISO/CEI 10746-3, lisez les articles 11 et 12 dans lesquels se trouvent une discussion sur l'emploi des normes dans la spécification des systèmes ODP et des exemples d'application des concepts ODP à la spécification des systèmes.

### 2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes internationales indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

#### 2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: le modèle de référence de base.*
- Recommandation UIT-T X.207 (1993) | ISO/CEI 9545:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de la couche application.*
- Recommandation UIT-T X.720 (1993) | ISO/CEI 10165-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: modèle d'information de gestion.*
- Recommandation UIT-T X.902 (1995) | ISO/CEI 10746-2:1996, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: fondements.*

- Recommandation UIT-T X.903 (1995) | ISO/CEI 10746-3:1996, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: architecture.*
- Recommandation UIT-T X.904 (1997) | ISO/CEI 10746-4:1998, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: sémantique architecturale.*

## 2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

- Recommandation UIT-T X.290 (1995), *Cadre général et méthodologie des tests de conformité d'interconnexion des systèmes ouverts pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Concepts généraux.*
- ISO/CEI 9646-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 1: Concepts généraux.*

## 2.3 Normes internationales

- ISO/CEI 11578-2:-<sup>1)</sup>: *Technologies de l'information – Interconnexion des systèmes ouverts – Appel de procédure distante – Partie 2: Notation de définition d'interface.*
- ISO/CEI TR 10000-1:1995, *Technologie de l'information – Cadre et taxonomie des profils normalisés internationaux – Partie 1: Principes généraux et cadre de documentation .*

## 3 Définitions

### 3.1 Définitions de la présente Recommandation | Norme internationale

Il n'y a pas de définitions dans la présente Recommandation | Norme internationale.

### 3.2 Définitions données dans d'autres Recommandations | Normes internationales

La présente Recommandation | Norme internationale emploie les termes suivants dont la définition se trouve dans la Rec. UIT-T X.902 | ISO/CEI 10746-2:

- abstraction;
- action;
- action interne;
- activité;
- affinement;
- architecture;
- atomicité;
- chaîne (d'actions);
- classe;
- classe de base;
- classe de gabarit;
- classe dérivée;
- communication;
- compatibilité;
- compatibilité de comportement;
- comportement de rattachement;
- comportement d'établissement;
- comportement de terminaison;
- comportement (d'un objet);
- composition;

---

1) A publier.

- configuration (d'objets);
- contexte contractuel;
- contrat;
- contrat d'environnement;
- courtage;
- création;
- décomposition;
- défaillance;
- défaut;
- détachement;
- domaine de désignation;
- données;
- environnement (d'objet);
- entité;
- erreur;
- état;
- fil d'exécution;
- gabarit d'action;
- identificateur;
- information;
- information de gestion;
- instance;
- instanciation;
- interdiction;
- interaction;
- interface;
- introduction (d'un <X>);
- invariant;
- liaison;
- nom;
- norme ODP;
- notification;
- objet;
- objet client;
- objet composite;
- objet consommateur;
- objet initiateur;
- objet producteur;
- objet répondeur;
- objet serveur;
- obligation;
- permission;
- persistance;
- point de conformité;
- point de référence;
- point de référence de programmation;

## ISO/CEI 10746-1 : 1998 (F)

- point de référence d'interfonctionnement;
- point de référence physique;
- point de vue;
- politique;
- portabilité;
- position dans l'espace;
- qualité de service;
- rattachement;
- résolution de nom;
- rôle;
- signature d'interface;
- sous-classe;
- sous-type;
- suppression;
- système;
- système ODP;
- transparence à la répartition;
- type;
- type-gabarit.

La présente Recommandation | Norme internationale emploie les termes suivants dont la définition se trouve dans la Rec. UIT-T X.903 | ISO/CEI 10746-3:

- action de liaison composite;
- action de liaison primitive;
- annonce;
- autorité de sécurité;
- cacher;
- canal;
- capsule;
- cible;
- communauté;
- désactivation;
- domaine de sécurité;
- fédération;
- flux;
- fonction ODP;
- gabarit de grappe;
- gestionnaire de capsule;
- gestionnaire de grappe;
- information de contrôle d'accès;
- interface de communication;
- interface flux;
- interface opération;
- interface signal;
- intercepteur;
- interrogation;
- invocation;

- langage <de point de vue>;
- liaison explicite;
- liaison implicite;
- migration;
- nœud;
- noyau;
- objet de liaison;
- objet protocole;
- objet d'ingénierie de base;
- objet lieu;
- point de reprise;
- point de vue entreprise;
- point de vue information;
- point de vue ingénierie;
- point de vue technologie;
- point de vue traitement;
- politique de sécurité;
- pose de point de reprise;
- réactivation;
- relocalisateur;
- reprise;
- schéma de duplication;
- schéma d'invariant;
- schéma dynamique;
- schéma statique;
- signal;
- signature d'interface opération;
- signature d'interface flux;
- signature d'interface signal;
- talon;
- terminaison;
- transparence à la duplication;
- transparence à la localisation;
- transparence à la migration;
- transparence à la persistance;
- transparence à la relocalisation;
- transparence aux défaillances;
- transparence aux transactions;
- transparence d'accès;
- valider.

La présente Recommandation | Norme internationale emploie les termes suivants dont la définition se trouve dans l'ISO/CEI 9646:

- déclaration de conformité d'une implémentation;
- informations complémentaires nécessaires aux essais de conformité d'une implémentation;
- point de contrôle et d'observation.

La présente Recommandation | Norme internationale emploie les termes suivants dont la définition se trouve dans la Rec. UIT-T X.200 | ISO/CEI 7498-1:

- système ouvert;
- syntaxe abstraite;
- syntaxe de transfert.

## 4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes sont utilisées:

A-profile	Profil d'application ( <i>application profile</i> )
ACID	Atomicité, cohérence, isolation, durabilité
AE	Entité d'application ( <i>application entity</i> )
AEI	Invocation d'entité d'application ( <i>application entity invocation</i> )
ALS	Structure de la couche Application ( <i>application layer structure</i> )
AP	Processus d'application ( <i>application process</i> )
API	Invocation de processus d'application ( <i>application process invocation</i> )
API	Interface de programme d'application ( <i>application program interface</i> )
ASO	Objet de service d'application ( <i>application service object</i> )
BEO	Objet d'ingénierie de base ( <i>basic engineering object</i> )
CAD	Conception assistée par ordinateur ( <i>computer aided design</i> )
CD	Disque compact ( <i>compact disk</i> )
CIM	Fabrication intégrée par ordinateur ( <i>computer integrated manufacturing</i> )
CMIP	Protocole commun d'informations de gestion ( <i>common management information protocol</i> )
CMIS	Service commun d'informations de gestion ( <i>common management information service</i> )
DL	Langage de définition ( <i>definition language</i> )
F-profile	Profil de format et de présentation ( <i>format and presentation profile</i> )
FDT	Technique de description formelle ( <i>formal description techniques</i> )
GUI	Interface graphique d'utilisateur ( <i>graphical user interface</i> )
HCI	Interface homme machine ( <i>human computer interface</i> )
ICS	Déclaration de conformité d'implémentation ( <i>implementation conformance statement</i> )
IDL	Langage de définition d'interface ( <i>interface definition language</i> )
IT	Techniques (ou Technologies) de l'information ( <i>information technology</i> )
IXIT	Informations complémentaires nécessaires aux essais de conformité d'une implémentation ( <i>implementation extra information for testing</i> )
MIM	Modèle d'informations de gestion ( <i>management information model</i> )
MMC(S)	(Système) de téléconférence multimédia [ <i>multimedia conferencing (system)</i> ]
ODP	Traitement réparti ouvert ( <i>open distributed processing</i> )
OMG	Groupe de gestion d'objet ( <i>object management group</i> )
OMT	Technique de modélisation par objets ( <i>object modelling technique</i> )
OSE	Environnement des systèmes ouverts ( <i>open system environment</i> )
OSF	Fondement des systèmes ouverts ( <i>open software foundation</i> )
OSI	Interconnexion des systèmes ouverts ( <i>open systems interconnection</i> )
PCO	Point de contrôle et d'observation ( <i>point of control and observation</i> )

QS	Qualité de service
RDA	Accès aux bases de données distantes ( <i>remote database access</i> )
RM-ODP	Modèle de référence pour le traitement réparti ouvert ( <i>reference model of open distributed processing</i> )
RPC	Appel de procédure distante ( <i>remote procedure call</i> )
T-profile	Profil de transport ( <i>transfer profile</i> )
TINA	Architecture de réseautage des informations de télécommunication ( <i>telecommunication information networking architecture</i> )
TVHD	Télévision haute définition
ULA	Architecture des couches supérieures ( <i>upper layers architecture</i> )

## 5 Conventions

La présente Recommandation | Norme internationale applique les conventions particulières suivantes:

- 1) lors de leur première apparition aux articles 7 et 8, les termes formels extraits des Rec. UIT-T X.902 | ISO/CEI 10746-2 et Rec. UIT-T X.903 | ISO/CEI 10746-3 sont mis en italique;
- 2) les exemples de l'article 12 utilisent les conventions de dessin OMT définies dans [Rumbaugh 91];
- 3) dans les diagrammes:
  - les objets sont représentés par des ovales ou des cercles;
  - le symbole "⊥" en saillie sur un objet représente une interface.

## 6 Normalisation ODP

### 6.1 Objectifs et motivations

La normalisation ODP a pour objectif d'établir des normes qui permettent de tirer avantage de la distribution de l'information dans un environnement caractérisé par la présence de moyens hétérogènes de traitement de l'information relevant de domaines d'organisation multiples. Ces normes traitent des contraintes à la spécification des systèmes et de la fourniture d'une infrastructure de système qui permettent de s'accommoder des difficultés inhérentes à la conception et à la programmation des systèmes répartis.

L'importance des systèmes répartis découle du besoin croissant d'interconnexion des systèmes de traitement de l'information. Ce besoin résulte de tendances d'ordre organisationnel qui, comme la réduction de taille, imposent des échanges d'informations tant entre groupes au sein d'une organisation donnée qu'entre organisations coopérantes. Les progrès de la technique permettent de répondre à ces tendances en donnant plus d'importance aux réseaux de services d'information et aux stations de travail personnelles et en permettant d'établir des applications réparties sur l'étendue de grands réseaux de systèmes interconnectés.

Pour être en mesure tant de gérer la répartition des systèmes que de l'exploiter, en tirant parti, par exemple, de son potentiel en termes de disponibilité, de performances, de sûreté de fonctionnement et d'optimisation des coûts, les organisations doivent être capables de s'accommoder d'un certain nombre de caractéristiques essentielles de la répartition des systèmes:

- **éloignement**: les composants d'un système réparti peuvent être dispersés dans l'espace; les interactions peuvent intervenir localement ou à distance;
- **concomitance**: tout composant d'un système réparti peut fonctionner en parallèle avec d'autres composants;
- **absence d'état global**: il n'est pas possible de déterminer avec précision l'état global d'un système réparti;
- **individualisation des défaillances**: tout composant d'un système réparti est susceptible de défaillance indépendamment de tout autre composant;
- **asynchronisme**: une horloge globale unique ne peut seule cadencer les activités de communication et de traitement informatique. Dans un système réparti, il n'est pas possible de supposer que des modifications corrélatives se produisent au même instant;

- **hétérogénéité**: il n'est pas garanti que la réalisation des composants d'un système réparti fasse appel aux mêmes solutions techniques, dont il est de plus certain qu'elles varieront dans le temps. L'hétérogénéité se manifeste en nombre d'endroits: matériels, systèmes d'exploitation, réseaux et protocoles de communication, langages de programmation, applications, etc.;
- **autonomie**: un système réparti peut s'étendre sur de nombreux domaines de gestion et de commande, régis par des autorités différentes, sans qu'il existe de centre de gestion unique. Le degré d'autonomie caractérise le point vers lequel des autorités distinctes exercent un commandement sur les ressources de traitement et sur les appareils qui leur sont associés, imprimantes, équipements de stockage de l'information, écrans graphiques, matériels audio, etc.;
- **évolutivité**: au cours de son existence opérationnelle, un système réparti devra en général faire face à de nombreuses modifications, motivées par des progrès techniques qui permettent d'améliorer les performances pour un coût moindre, par des décisions stratégiques relatives à de nouveaux objectifs et par de nouveaux types d'applications;
- **mobilité**: les sources d'information, les nœuds de traitement et les utilisateurs sont susceptibles de mobilité physique. Les programmes et les données peuvent aussi être déplacés de nœud en nœud pour, par exemple, tenir compte de la mobilité physique ou pour optimiser les performances.

Il n'est pas facile de construire des systèmes de cette sorte. Il y faut une architecture mais, comme une réalisation unique est incapable de répondre à toutes les exigences, cette architecture doit faire preuve de souplesse. Qui plus est, comme un fournisseur unique ne sera pas en mesure d'offrir toutes les solutions, il est essentiel que l'architecture et les fonctions nécessaires à sa réalisation soient définies sous forme d'un ensemble de normes grâce auxquelles de nombreux fournisseurs pourront collaborer à l'élaboration des systèmes répartis. Ces normes permettront de construire des systèmes qui pourront:

- être **ouverts** – en offrant à la fois portabilité (exécution des composants sur des nœuds de traitement différents sans modification de ces composants) et interfonctionnement (interactions significatives entre composants résidant éventuellement dans des systèmes différents);
- être **intégrés** – en incorporant dans un ensemble unique une variété de systèmes et de ressources sans nécessiter de coûteux développements spécifiques. Il peut s'agir de systèmes d'architectures différentes, de ressources différentes de performances différentes. L'intégration aide à la prise en compte de l'hétérogénéité;
- être **souples** – en étant capables à la fois d'évoluer et de prendre en compte le fait que les systèmes existants doivent continuer à fonctionner. Un système ouvert devrait pouvoir faire face à des changements qui, comme une reconfiguration dynamique rendue nécessaire par l'évolution de la situation, peuvent survenir pendant l'exécution. La souplesse aide à la prise en compte de la mobilité;
- être **modulaires** – en octroyant aux parties du système l'autonomie dans le maintien de leurs relations. La modularité est à la base de la souplesse;
- être **fédérables** – en permettant de combiner des systèmes appartenant à différents domaines administratifs ou techniques pour atteindre un objectif unique;
- être **gérables** – en permettant de contrôler, de commander et de gérer les ressources du système afin d'assurer la prise en charge des politiques de configuration, de qualité de service et de suivi comptable;
- répondre aux besoins de **qualité de service** – en faisant preuve, par exemple, d'aptitudes en matière d'opportunité temporelle, de disponibilité et de fiabilité dans un contexte de ressources distantes et d'interactions à distance, en même temps que de résistance aux pannes, propriété grâce à laquelle la défaillance d'une partie d'un système réparti n'empêche pas le fonctionnement du reste de ce système. Savoir résister aux pannes, et plus généralement fonctionner de manière sûre, sont des qualités nécessaires aux grands systèmes répartis, car il est peu probable que toutes leurs parties puissent jamais être simultanément opérationnelles;
- assurer la **sécurité** – en garantissant que les services offerts par le système et les données qui lui sont confiées sont protégés contre des accès non autorisés. Il est plus difficile d'assurer le niveau de sécurité requis dès lors que les interactions ont lieu à distance et que des parties du système et ses utilisateurs sont mobiles;

- offrir la **transparence** – en cachant aux applications les détails et les différences entre les mécanismes qui servent à surmonter les problèmes causés par la répartition. Il s'agit là d'une exigence fondamentale, qui découle du besoin de faciliter l'élaboration d'applications réparties. Parmi les aspects relatifs à la répartition dont il convient de masquer totalement ou partiellement les effets, se trouvent: l'hétérogénéité des matériels et des logiciels sous-jacents, la localisation et la mobilité des composants, les mécanismes qui permettent d'atteindre le niveau requis de qualité de service en cas de défaillance (par exemple, duplication, migration, pose de points de reprise, etc.).

## 6.2 Réalisation

La normalisation ODP comporte quatre éléments fondamentaux:

- une approche à la spécification des systèmes reposant sur une modélisation par objets;
- la spécification d'un système par la spécification de points de vue distincts mais apparentés;
- la définition d'une infrastructure qui apporte aux applications du système les transparences à la répartition;
- un cadre d'évaluation de la conformité des systèmes.

### 6.2.1 Modélisation par objets

La modélisation par objets formalise les pratiques de conception bien établies que sont l'abstraction et l'encapsulation. L'abstraction permet d'isoler la description des fonctions du système de la spécification des détails de mise en œuvre. L'encapsulation permet de cacher l'hétérogénéité, de localiser les défaillances, de mettre la sécurité en œuvre et de dissimuler à leurs utilisateurs les mécanismes de fourniture des services.

Les concepts de modélisation par objets se composent:

- de concepts de modélisation de base, regroupant un minimum de concepts rigoureusement définis (action, objet, interaction et interface) sur lesquels se fondent les descriptions des systèmes ODP et qui sont applicables dans tous les points de vue;
- de concepts de spécification, traitant de notions qui, comme types et classes, sont nécessaires pour pouvoir raisonner sur les spécifications et sur les relations entre spécifications, offrant des outils généraux de conception et établissant les exigences portant sur les langages de spécification;
- de concepts de structuration, bâtissant sur les concepts de modélisation et de spécification pour s'intéresser aux structures récurrentes des systèmes répartis et traitant des questions de politique, de désignation, de comportement, de sûreté de fonctionnement et de communication.

### 6.2.2 Spécifications de points de vue

Un point de vue (sur un système) est une abstraction qui produit une spécification de la totalité du système par rapport à un ensemble particulier de considérations. Il a été choisi cinq points de vue qui couvrent, simplement mais complètement, tous les domaines de la conception architecturale. Ce sont:

- le point de vue entreprise, qui s'occupe des buts, du champ d'action et des politiques qui gouvernent les activités du système spécifié au sein de l'organisation dont il fait partie;
- le point de vue information, qui s'occupe des types d'information que manipule le système et des contraintes qui pèsent sur l'usage et l'interprétation faite de cette information;
- le point de vue traitement, qui s'occupe de la décomposition fonctionnelle du système en un ensemble d'objets qui interagissent à des interfaces pour permettre de répartir le système;
- le point de vue ingénierie, qui s'occupe de l'infrastructure nécessaire pour prendre en charge la répartition du système;
- le point de vue technologie, qui s'occupe des choix de solutions techniques capables de prendre en charge la répartition du système.

A chaque point de vue est associé un langage de point de vue qu'il est possible d'utiliser pour exprimer une spécification du système sous ce point de vue. Les concepts de modélisation par objets fournissent une base commune aux langages de points de vue et rendent possible l'identification des relations entre les différentes spécifications de points de vue et l'expression de correspondances entre les représentations du système sous les différents points de vue.

### 6.2.3 Transparences à la répartition

Les transparences à la répartition permettent de cacher aux applications les complexités qui résultent de la répartition lorsqu'elles ne les concernent pas. Par exemple:

- la transparence d'accès masque les différences entre systèmes dans l'ordre de la représentation des données et des mécanismes d'appel des services;
- la transparence à la localisation évite à une application d'avoir à connaître la localisation d'un service auquel elle doit faire appel;
- la transparence à la relocalisation cache à une application un changement dans la localisation d'un service qu'elle utilise;
- la transparence à la duplication masque le fait qu'il peut exister, aux fins de fiabilité et de disponibilité, plusieurs copies d'un même service.

Les normes ODP définissent les fonctions et les structures nécessaires à la réalisation des transparences à la distribution. A chaque transparence, pourtant, il faut associer la recherche du juste équilibre entre performance et coût. Dans de nombreux cas, de plus, seules seront pertinentes des transparences particulières. Un système ODP doit donc mettre en œuvre, dans le respect des normes afférentes, celles des transparences qu'il prend en charge, mais il n'a pas l'obligation de les prendre toutes en charge.

### 6.2.4 Conformité

Une conséquence des propriétés fondamentales d'hétérogénéité et d'évolutivité est qu'il est possible de se procurer séparément les différentes parties d'un système, auprès de fournisseurs différents. Il est donc très important que le comportement des différentes parties d'un système soit clairement défini et qu'il soit possible de désigner les responsables en cas de manquement du système à tenir ses spécifications.

Le cadre défini pour régir l'expression de la conformité traite de ces questions. Il comporte:

- la détermination, au sein de l'ensemble des spécifications de points de vue, de points de conformité où il est possible d'effectuer des observations sur la conformité;
- la définition de classes de points de conformité;
- la spécification de la nature des déclarations de conformité qu'il est possible de faire dans chaque point de vue et des relations entre elles.

## 6.3 Normes

### 6.3.1 Le modèle de référence

Le modèle de référence ODP donne le cadre général dans lequel se situe la normalisation ODP. Il se compose de deux parties principales:

- la Rec. UIT-T X.902 | ISO/CEI 10746-2: **fondements**, qui contient la définition des concepts et le cadre analytique à utiliser pour décrire les systèmes de traitement réparti et qui introduit les principes concernant l'évaluation de la conformité aux normes ODP;
- la Rec. UIT-T X.903 | ISO/CEI 10746-3: **architecture**, qui contient la spécification des caractéristiques d'un système ODP et de l'infrastructure qui apporte les transparences à la répartition.

La Rec. UIT-T X.904 | ISO/CEI 10746-4: **sémantique d'architecture** complète ces deux parties principales par l'apport d'une interprétation formelle des concepts de modélisation et des langages de points de vue à partir de techniques existantes de descriptions formelles.

Le modèle de référence ODP est de nature générique, c'est-à-dire qu'il ne dépend d'aucun des domaines d'application qui font appel aux techniques de systèmes répartis. Il reste donc applicable à n'importe lequel de ces domaines. Dans le cas de certains domaines d'application particuliers, il y aura lieu de raffiner et de spécialiser le modèle de référence ODP pour l'adapter à des besoins spécifiques. Il en résultera:

- des modèles de référence spéciaux adaptés à certains types d'entreprises. Ils utilisent les concepts et les fonctions communes donnés par le modèle de référence ODP, complétés par des détails conceptuels additionnels et des fonctions spécifiques. Tel est le cas, par exemple, de TINA (*telecommunication information networking architecture*);
- des normes pour la spécification des fonctions spéciales requises par des applications particulières. Tel est le cas, par exemple, des interfaces d'appels téléphoniques. Ces fonctions pourront avoir été signalées dans un modèle de référence spécial.

Une conséquence de sa généralité est que le modèle de référence ODP permet, pour bâtir des solutions qui répondent aux exigences exprimées dans les affaires, d'intégrer d'une manière économique, mais efficace, des techniques disparates de systèmes répartis. C'est ainsi que dans le cas des architectures publiées par l'OSF (*open software foundation*) et par l'OMG (*object management group*), l'ODP, parce qu'il traite des questions de fédération, de transparence et de gestion-systèmes et qu'il définit un réseau dense de points de référence où s'intègrent des fonctions d'origines diverses, permet de mieux expliquer les raisons pour lesquelles les fonctions spécifiées s'accordent pour prendre correctement en charge des systèmes répartis.

### 6.3.2 Normes spécifiques

On peut distinguer quatre catégories de normes au sein du cadre général qu'apporte le modèle de référence ODP:

- des cadres architecturaux additionnels, qui complètent le modèle de référence ODP dans des domaines spéciaux tels que désignation, sécurité et évaluation de la conformité;
- des normes de notation, qui définissent les notations à utiliser pour exprimer les spécifications des différents aspects de l'intégration et de la répartition des systèmes et les règles d'expression des relations entre les différentes spécifications;
- des normes de composants, qui définissent chacune, soit une fonction ODP particulière, soit un groupe de fonctions étroitement apparentées susceptibles de réalisation sous forme de plate-forme matérielle ou logicielle unique;
- des normes d'assemblage de composants, qui définissent comment coordonner un certain nombre de composants pour atteindre un but particulier que vise l'ensemble du système. La réalisation d'une transparence particulière donne un exemple d'un tel but.

NOTE – Il peut arriver que des normes spécifient aussi bien les composants que leur assemblage, conduisant ainsi à la réalisation directe d'une fonction utile pour les systèmes répartis. D'autres normes peuvent servir de fondement à plusieurs normes d'assemblage de composants. Des normes d'assemblage de composants visant à apporter les transparences à la localisation ou à la migration pourraient, par exemple, faire référence à une norme de relocalisateur ODP.

Le cadre dans lequel le modèle de référence ODP situe les normes de composants et d'assemblage de composants en permet la mise en œuvre par diverses voies. Cette souplesse est garantie d'une raisonnable longévité du cadre car elle autorise la prise en compte de nouvelles solutions au fur et à mesure de leur arrivée à maturité. C'est ainsi qu'une norme spécifique, ou un groupe de normes, peut contenir la spécification d'une solution particulière qui répond à une exigence ODP donnée tout en laissant la liberté de choix particuliers pour la réalisation de produits ouverts. Il peut exister plusieurs normes de ce genre, correspondant à différentes approches. Au cours du temps, l'incorporation de nouvelles solutions techniques conduira à de nouvelles générations de normes au sein du même cadre ODP.

## 7 Fondements

La Rec. UIT-T X.902 | ISO/CEI 10746-2 définit un ensemble de concepts de modélisation sur lesquels se fonde l'expression de l'*architecture* des *systèmes ODP* définis dans la Rec. UIT-T X.903 | ISO/CEI 10746-3. Ces concepts se regroupent en trois catégories:

- les concepts de modélisation de base, qui présentent un modèle général par objets. D'une manière générale, il est possible de décrire un système ODP sous la forme d'une collection d'*objets* reliés entre eux et en interaction mutuelle;
- les concepts de spécification, qui ne font pas partie intégrante des systèmes répartis, mais grâce auxquels leur utilisateur peut en décrire les spécifications et raisonner à leur sujet. Ils posent des exigences portant sur tout langage de spécification employé pour spécifier un système ODP. Comme ils sont essentiellement indépendants des langages, ils sont applicables dans le cadre de n'importe quel langage de spécification ou de programmation;
- les concepts de structuration, qui concernent l'organisation, les propriétés des systèmes et des objets, *politique*, *désignation*, *comportement* et gestion, correspondant à des notions et à des structures d'application générale dans la conception et la description des systèmes répartis.

La Rec. UIT-T X.902 | ISO/CEI 10746-2 apporte aussi un cadre général, exprimé selon les termes du modèle général par objets, qui permet d'appréhender les questions de *conformité* dans les systèmes ODP. Ce cadre est présenté à l'article 9.

## 7.1 Concepts de modélisation de base

### 7.1.1 Objets

Les spécifications des systèmes ODP sont exprimées en termes d'objets. Un objet est une représentation d'une *entité* du monde réel. Il contient de l'*information* et offre des services. Un système se compose d'objets en interaction mutuelle. Un objet se caractérise par ce qui le distingue de tout autre objet et par ses aspects en termes d'*encapsulation*, d'*abstraction* et de comportement.

L'encapsulation est la propriété que possède l'information contenue dans un objet de n'être accessible qu'au travers d'*interactions* qui se produisent aux *interfaces* que l'objet présente. Du fait de l'encapsulation, les interactions ne créent pas d'effets annexes. Ceci signifie qu'une interaction avec un objet ne peut affecter l'*état* d'un autre objet sans qu'intervienne une interaction secondaire avec cet objet. Par conséquent, toute transition d'état d'un objet ne peut résulter que d'une *action interne* à l'objet ou d'une interaction de l'objet avec son *environnement*.

L'abstraction implique que les détails internes d'un objet sont cachés aux autres objets. Son rôle est crucial à l'égard de l'hétérogénéité car elle permet de mettre en œuvre différents services de différentes manières par emploi de mécanismes et de solutions techniques différents. C'est la base de la *portabilité* et de l'interopérabilité.

De plus, l'abstraction isole complètement les objets, ce qui permet de les remplacer ou de les modifier sans en changer l'environnement, pourvu qu'ils continuent à offrir les services que leur environnement attend d'eux, manifestant donc une capacité de compatibilité arrière. C'est une approche à la notion d'extensibilité, si essentielle dans les grands environnements répartis hétérogènes qui, de par leur nature même, sont en évolution constante. Un modèle par objets est modulaire et donne les moyens de composer de nouveaux modules à partir de modules existants. Ce sont là des caractéristiques d'importance si l'on veut construire des systèmes évolutifs et pousser au réemploi des logiciels dans un souci d'amélioration de la productivité.

Le modèle de référence ODP, qui est général, part d'un très petit nombre d'hypothèses. Par exemple:

- le niveau de granularité des objets est arbitraire: ils peuvent, par exemple, être aussi vastes que le réseau téléphonique ou aussi petits qu'un nombre entier;
- les objets peuvent afficher des comportements encapsulés arbitraires et être doués d'un niveau arbitraire de parallélisme interne;
- il n'existe pas de contraintes portant sur les interactions entre objets. On peut trouver, par exemple, des interactions asynchrones et synchrones multiparties.

### 7.1.2 Interfaces et points d'interaction

Les objets ne peuvent agir les uns sur les autres que par des interfaces. Une interface représente une partie du comportement de l'objet, relative à un sous-ensemble donné des interactions auxquelles il peut participer. Chaque interface est identifiable à un ensemble d'interactions auxquelles l'objet peut participer. Il est à noter que ces interactions ne prennent pas nécessairement place avec d'autres objets: un objet peut avoir des interactions avec lui-même. Une importante caractéristique des objets du modèle de référence ODP est qu'un objet peut disposer de plusieurs interfaces. Des considérations de séparation fonctionnelle et de répartition en donnent les raisons. La séparation fonctionnelle peut se comprendre, par exemple, dans le contexte de la gestion-systèmes, où les interactions de gestion et les interactions qui ne ressortissent pas à la gestion se trouvent normalement divisées entre des interfaces différentes. Lorsque l'on considère des objets répartis, la séparation s'avère aussi nécessaire là où les interfaces qui servent de points d'accès à l'objet sont situées à des emplacements différents dans l'espace.

Munir un objet de plusieurs interfaces a pour conséquence que les interactions qui se produisent sur une interface peuvent être influencées par celles qui se déroulent sur d'autres interfaces. Elles ne sont donc pas nécessairement déterminées isolément les unes des autres.

Une interface existe en un *point d'interaction* qui, à tout instant dans le temps, se trouve associé à un certain point dans l'espace. Plusieurs interfaces peuvent exister à un point d'interaction donné. Le point d'interaction peut être doué de mobilité. L'affectation des points dans l'espace et dans le temps et la manière de l'exprimer dépendent du langage dans lequel est exprimée la spécification.

### 7.1.3 Comportement et état

Le comportement d'un objet est une collection d'*actions* auxquelles peut prendre part cet objet, associée au jeu de contraintes qui portent sur les circonstances dans lesquelles ces actions peuvent se produire. Le modèle par objets n'impose pas de contraintes quant à la forme ou à la nature du comportement de l'objet. Les actions peuvent être soit des interactions entre l'objet et son environnement, soit des actions internes.

Les concepts d'état et de comportement sont liés. L'état d'un objet est la condition de cet objet à un instant donné. Il détermine les séquences d'actions dans lesquelles il est possible que l'objet soit impliqué. En même temps, les actions induisent des transitions d'état. Par conséquent, l'état d'un objet est déterminé en partie par son comportement passé. Les actions qu'un objet va effectivement entreprendre ne sont naturellement pas dictées par son seul état présent, mais vont dépendre aussi des actions auxquelles son environnement est en mesure de participer.

## 7.2 Concepts de spécification

### 7.2.1 Composition et décomposition

Il est possible d'utiliser les concepts de *composition* et de *décomposition* pour organiser la spécification d'un système réparti sous forme d'un ensemble de spécifications dont chacune traite d'un niveau différent d'abstraction. La spécification d'un système réparti complexe peut alors se ramener à la spécification d'un certain nombre d'objets plus simples, décomposables à leur tour à un niveau d'abstraction plus bas.

Les processus de composition et de décomposition donnent les moyens de procéder à la spécification hiérarchique d'une application répartie. Dans une telle hiérarchie de composition, les *classes* d'objets des niveaux supérieurs sont assemblées à partir de *configurations* de classes d'objets constitutives de niveau inférieur. La composition est donc un puissant concept de modélisation puisqu'elle permet de traiter un sous-système comme un objet unique de niveau supérieur.

A titre d'exemple, le schéma de la Figure 1 représente la composition de deux objets, C et D. Pour pouvoir composer C avec D, il faut que le comportement de C soit défini de telle sorte qu'il interagisse avec D de manière appropriée. L'interface entre C et D peut avoir la simplicité d'une interaction unique (qui ferait passer, par exemple, de l'information de C à D), ou être plus compliquée (comme dans le cas d'une séquence d'interactions au cours de laquelle C appellerait une opération qui renvoie un résultat). Si C et D se voient composés en un objet unique, alors les interactions entre C et D se trouvent cachées et deviennent des actions internes à l'*objet composite*. L'interface de gauche représente une interface de l'objet composite.

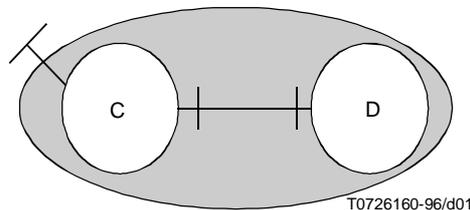


Figure 1 – Composition d'objets

La composition des objets conduit à la composition des états et des comportements. Il est donc possible de parler de comportement composite et d'état composite.

La hiérarchie de composition est orthogonale à la hiérarchie de *sous-classes*, étudiée au 7.2.3, avec laquelle elle ne doit pas être confondue. Il n'existe pas en général de relation de hiérarchie de sous-classes entre les classes des objets composants et les classes de l'objet composite. Il peut même se trouver des exemples où l'objet composite appartient à une sous-classe de la classe à laquelle appartient un objet composant (par exemple, lorsque les objets représentent des services de communication à l'un desquels l'objet composite ajoute de la valeur).

### 7.2.2 Compatibilité de comportement

Un objet est dit présenter une compatibilité de comportement avec un autre objet dans un environnement donné si le premier objet peut prendre la place du second sans que l'environnement soit capable de détecter la différence. Toute interprétation particulière de *compatibilité de comportement* imposera des contraintes sur le comportement permis à l'environnement. Une approche usuelle consiste à faire l'hypothèse que l'environnement se comporte comme un testeur de l'objet initial. En d'autres termes, l'environnement devrait être capable d'exercer le comportement initial dans sa totalité, mais sans pouvoir en faire plus. Il est essentiel de formuler ces hypothèses dès lors que la compatibilité de comportement est à envisager dans le contexte d'un environnement inconnu.

### 7.2.3 Type et classe

Un *type* est un prédicat, c'est-à-dire une propriété, ou un ensemble de propriétés, portant sur une collection de choses (objets, interfaces, etc.). Par exemple, "est rouge" est un type. On dit que quelque chose satisfait un type, ou est du type, si le prédicat tient pour la chose. Les choses peuvent être très différentes, tout en satisfaisant le même type; il suffit qu'elles possèdent les propriétés que prescrit le type. Par exemple, un drapeau donné, une maison de briques donnée et une voiture de sport donnée peuvent tous être rouges.

Les types organisent implicitement les choses sous forme d'ensembles appelés classes. Une classe est la collection des choses qui présentent les propriétés décrites par le type.

La notion de type est très générale. Il est possible de la spécialiser de diverses manières. Elle est utile dans tout contexte où il est nécessaire de parler des propriétés des choses, de raisonner à leur sujet et de les vérifier (aux fins, par exemple, de *courtage*, ou de *liaison*).

Des concepts de type et de classe découlent des hiérarchies naturelles, de classe à sous-classe et de type à sous-type. La distinction entre classe et sous-classe correspond à la distinction intuitive que fait la théorie des ensembles entre ensemble et sous-ensemble. Une classe est sous-classe d'une autre si et seulement si la première est un sous-ensemble de la dernière. Un type est sous-type d'un autre si le prédicat du premier implique le prédicat du second.

Les notions de sous-classes et de sous-types vont de pair. A chaque type correspond une classe associée, qui peut naturellement être vide. Si donc nous avons deux types, T1 et T2, il doit exister des classes C1 et C2 qui leur sont associées. T1 est sous-type de T2 précisément quand C1 est sous-classe de C2.

### 7.2.4 Gabarits

Un *gabarit* décrit une collection de choses (objets, interfaces, etc.) à un niveau de détail tel qu'il suffise pour pouvoir créer une nouvelle instance de chose.

Pour décrire un ensemble d'objets, le gabarit en donne des caractéristiques telles que paramètres d'état, opérations et comportement. L'*instanciation* d'un objet impose normalement d'en établir l'état initial. On pourrait par exemple créer un objet mémoire tampon de contenu vide.

Le concept de compatibilité de comportement s'applique aussi aux gabarits d'objets en ce sens qu'il y a compatibilité de comportement entre deux gabarits d'objets si les objets instanciés à partir de ces gabarits présentent la compatibilité de comportement.

Un *type-gabarit* est un prédicat défini dans un gabarit. Un type-gabarit est satisfait par toute instanciation issue du gabarit. Il peut aussi, de manière générale, être satisfait par d'autres choses, du moment qu'elles satisfont aux mêmes besoins que les instanciations. Il est possible, par exemple, de définir un type-gabarit de telle sorte que des objets instanciés à partir de gabarits différents, mais satisfaisant ce type-gabarit, présentent une compatibilité de comportement.

Chaque type-gabarit donne naissance à une *classe de gabarits*: l'ensemble des *instances* du type-gabarit. Les classes de gabarits peuvent être organisées selon des hiérarchies de sous-classes, conséquences des relations entre types et sous-types qui existent entre les types-gabarits.

### 7.2.5 Rôles

Dans le cadre d'un gabarit d'objet composite, un *rôle* détermine un comportement à associer à l'un des objets composants.

Un rôle peut correspondre à un sous-ensemble du comportement total d'un objet composite. Lorsqu'on considère un objet du point de vue d'un rôle, seul importe un sous-ensemble nommé de ses actions. Ses autres actions sont rejetées dans l'abstraction et éventuellement reprises par d'autres rôles. En fonction de ses interactions, un objet composite peut jouer plusieurs rôles à un instant donné; il peut jouer des rôles différents à des instants différents. Ces rôles peuvent être associés à des interfaces.

Un objet peut par exemple tenir son rôle fonctionnel normal, qui correspond à la mission pour laquelle il a été conçu, tout en jouant, dans le cadre de la gestion, un rôle de gestion dans lequel il se comporte de manière à contrôler et à commander son comportement dans le rôle correspondant à sa mission. Chaque rôle est muni de sa propre interface: le rôle lié à la mission est associé à une interface de mission et le rôle lié à la gestion est associé à une interface de gestion.

### 7.2.6 Classes de base et classes dérivées

Les concepts de *classe de base* et de *classe dérivée* sont fondés sur une notion générale, celle d'une modification des gabarits connue sous le nom de *modification incrémentale*. Il s'agit, par modification d'un gabarit existant, d'en dériver un nouveau gabarit. Le nouveau gabarit s'appelle gabarit dérivé et le gabarit d'origine est le gabarit de base. Les instances du gabarit de base forment la classe de base et celles du gabarit dérivé la classe dérivée.

La modification incrémentale, qui permet la substitution, conduit en général à une hiérarchie différente de celle des classes et sous-classes.

Soit par exemple une classe de gabarit C1 constituée de voitures rouges qui soit définie par un gabarit Temp1 contenant la ligne:

COULEUR=ROUGE.

Supposons que pour faire le gabarit C2 cette ligne soit remplacée par:

COULEUR=BLEU.

La classe de gabarit C2 est dérivée de C1 mais n'en constitue pas une sous-classe.

Dans certains cas, la mise en œuvre d'une classe dérivée peut être fondée sur la mise en œuvre de la classe de base. C'est le concept d'héritage de mise en œuvre, qui permet à des programmes exécutables de partager du code entre eux. Mais ceci peut être source de difficultés dans un environnement réparti, car il est nécessaire de propager chaque modification apportée au code de la classe de base à toutes les mises en œuvre de classes dérivées. Il n'est donc pas exigé des systèmes ODP qu'ils prennent en charge l'héritage de mise en œuvre.

## 7.3 Concepts de structuration

### 7.3.1 Groupes et domaines

Un *groupe* est un ensemble d'objets rassemblés pour des raisons structurelles ou parce que leur comportement présente des aspects communs (par exemple, les objets d'un groupe de duplication peuvent se remplacer l'un l'autre, ceux d'un groupe de communication peuvent participer à la même interaction). Comme il est général, le concept de groupe permet de spécifier différentes espèces de groupes, que les systèmes répartis utilisent à des fins variées, telles que résistance aux *défaillances*, disponibilité et, par exemple dans le cas des applications de téléconférence, prise en charge des applications.

Un *domaine* est une forme particulière de groupe, dans laquelle un aspect particulier du comportement des objets du groupe est sous le contrôle d'une même autorité. Dans un *domaine de sécurité* par exemple, les politiques de sécurité qui s'appliquent aux objets du domaine sont établies par la même *autorité de sécurité*. Le concept de domaine permet d'introduire dans les systèmes répartis les notions d'autonomie, d'autorité et de commande. Il répond à un grand nombre de besoins divers car les systèmes répartis font appel à de nombreuses espèces de domaines (*domaines de désignation*, de sécurité, ou de gestion, par exemple).

### 7.3.2 Désignation

Il est essentiel de savoir désigner les composants d'un système réparti pour les distinguer les uns des autres et pour y avoir accès. La fonction de désignation constitue donc un élément fondamental de la construction d'un système réparti.

Pour pouvoir traiter des questions d'hétérogénéité, d'autonomie et de *fédération*, il est nécessaire que la désignation et la gestion des *noms* tiennent compte du contexte. De là découle la possibilité, source de souplesse et d'évolutivité, de construire des systèmes de désignation indépendants, de les combiner de manière arbitraire et même, d'y faire participer des systèmes indépendants existants. C'est aussi la possibilité d'introduire de l'hétérogénéité entre différents niveaux des systèmes de désignation: des systèmes différents peuvent par exemple différer par l'emploi des formes de noms, la gestion de l'allocation des noms, l'application de politiques de désignation ou de stratégies de *résolution de noms*.

Les noms sont utilisés pour faire référence, dans un contexte donné, à des entités. Un nom peut dans certains cas se référer à plusieurs entités. Un nom qui se réfère sans ambiguïté à une seule entité est appelé *identifiant ou identificateur*.

Les concepts de désignation qui figurent dans la Rec. UIT-T X.902 | ISO/CEI 10746-2 ne spécifient pas la totalité d'un cadre de désignation pour l'ODP. Ceci fait l'objet d'une normalisation à part.

### 7.3.3 Contrat

Un *contrat* est un accord qui régit la coopération entre un certain nombre d'objets. Il emporte des idées d'*obligation*, de *permission*, d'*interdiction* et d'*attente* associées aux objets coopérants. Il s'agit donc là d'un concept servant à caractériser et à réglementer la coopération entre les objets.

Dès lors que des objets coopèrent et donc interagissent, il existe un certain contrat entre eux. Dans les cas où le contrat est passé à un moment donné pour se terminer plus tard, il s'agit d'une spécification dynamique de la configuration des objets. Bien que soit maintenue constante la potentialité de coopération entre les objets, les règles fournies par le contrat restreignent cette potentialité à ne s'exercer que pendant la durée d'un comportement provisoire.

Il arrive cependant souvent que les contrats découlent de règles exprimées dans le *langage de point de vue* sans nécessiter de déclaration explicite. Dans le cas par exemple du langage de traitement, un client n'est pas autorisé à faire appel à une opération qui ne serait pas définie sur l'interface du serveur. Seuls ont à être exprimés explicitement les contrats qui spécifient des contraintes supplémentaires.

Un contrat peut spécifier par exemple:

- les rôles tenus par les objets et les obligations qui portent sur ces rôles, c'est-à-dire le comportement attendu de leur coopération;
- les aspects *qualité de service* de la coopération entre les objets, tels que les questions de sûreté de fonctionnement, d'exactitude, etc.;
- le type de comportement qui *invalide* le contrat.

Un *contrat d'environnement* est un type particulier de contrat passé entre un objet et son environnement. Il explicite ce que l'objet exige de son environnement et réciproquement. Il se préoccupe en particulier des contraintes portant sur la qualité de service.

### 7.3.4 Liaison et relation de liaison

Un *comportement de liaison* établit un *contexte contractuel* (une liaison) entre des interfaces et permet aux objets de coopérer. Il peut exister des liaisons à plusieurs niveaux d'abstraction.

Une *relation de liaison* est la relation qui existe entre les objets qui coopèrent sous l'égide d'une liaison. Lorsque la relation de liaison est établie, l'objet sait que les autres objets qui participent à la relation de liaison obéissent au contrat. Un objet peut être impliqué simultanément dans plusieurs relations de liaison à chacune desquelles correspond un contrat.

NOTE – L'exemple du prochain article fait appel aux termes suivants, qui appartiennent à la terminologie de l'OSI: Bind, Bind Response, Unbind et contexte d'application.

Le comportement d'établissement d'une association en OSI donne un exemple d'établissement de liaison. Le *comportement d'établissement* est fourni par une opération Bind au cours de laquelle l'objet d'application initiateur envoie à l'objet d'application répondeur les paramètres du contrat. *L'objet en réponse* renvoie, pour établir la liaison, une opération de Bind Response qui comporte éventuellement des paramètres différents. Le contexte contractuel qui prévaut sur la liaison est donné par le contexte d'application sur lequel s'est fait l'accord par l'échange Bind et Bind Response. Lorsque l'un des partenaires souhaite terminer la liaison, il lance le *comportement de terminaison* en faisant appel à l'opération Unbind.

## 8 Architecture

La Rec. UIT-T X.903 | ISO/CEI 10746-3 formule les prescriptions normatives auxquelles doit se soumettre tout système pour pouvoir prétendre à la qualification ODP. Sur la base de la terminologie et des concepts définis dans la Rec. UIT-T X.902 | ISO/CEI 10746-2, elle définit:

- un cadre architectural, dans lequel la spécification des systèmes ODP est structurée par application des concepts de *points de vue*, de *spécifications de point de vue* et de *transparences à la répartition*;
- un ensemble de *langages*, dans les termes desquels peuvent s'exprimer les différentes spécifications de points de vue;
- une infrastructure de système, qui apporte aux applications les *transparences à la répartition*.

### 8.1 Cadre architectural

Il peut advenir que les systèmes répartis soient vastes et compliqués. Les nombreuses considérations différentes qui influent sur leur conception peuvent se résoudre en un corps de spécifications très substantiel dont la gestion, pour être un succès, demande qu'il soit bien organisé. Un bon cadre architectural serait celui qui permettrait de travailler séparément sur les parties indépendantes du projet, tout en donnant les moyens de déterminer clairement les contraintes mutuelles qu'imposent leurs différents aspects. Deux approches structurantes sont utilisées à cette fin dans l'architecture ODP, les points de vue et les transparences.

### 8.1.1 Points de vue

Un point de vue est une subdivision de la spécification d'un système complet conçue pour rassembler ceux des éléments d'information qui relèvent d'un domaine sur lequel porte un effort particulier au cours de la conception du système. Est système ODP tout ce qui présente de l'intérêt. Il peut donc s'agir aussi bien du système de traitement de l'information d'une entreprise que d'un élément matériel ou logiciel particulier d'un tel système. Les points de vue ne sont pas complètement indépendants les uns des autres: on trouve dans chacun d'eux des éléments essentiels qui sont apparentés à des éléments appartenant à d'autres points de vue. Ils restent cependant suffisamment indépendants pour qu'en puisse être simplifié le raisonnement qui conduit à la spécification complète.

Chacun des points de vue est en relation avec tous les autres. Ils ne forment pas une séquence rigide comme celle des couches de protocoles de communication, ni ne sont créés dans un ordre déterminé par quelque méthodologie de conception. L'architecture s'exprime sous la forme du jeu complet des points de vue apparentés, sans que soit posée de règle sur la manière d'établir la spécification complète d'un système donné.

Le modèle de référence ODP définit cinq points de vue qui sont:

- a) le *point de vue entreprise*: un point de vue sur le système et son environnement qui s'intéresse aux objectifs, au domaine d'application et aux politiques afférentes au système;
- b) le *point de vue information*: un point de vue sur le système et son environnement qui s'intéresse à la sémantique de l'information et au traitement qu'elle subit;
- c) le *point de vue traitement*: un point de vue sur le système et son environnement qui s'intéresse à la répartition par décomposition fonctionnelle du système en objets exerçant à leurs interfaces des interactions mutuelles;
- d) le *point de vue ingénierie*: un point de vue sur le système et son environnement qui s'intéresse aux mécanismes et aux fonctions nécessaires pour mettre en œuvre les interactions réparties entre objets du système;
- e) le *point de vue technologie*: un point de vue sur le système et son environnement qui s'intéresse aux choix de solutions techniques dans ce système.

Afin de représenter un système ODP sous un point de vue donné, il faut définir un ensemble structuré de concepts dans les termes desquels il sera possible d'exprimer cette représentation (ou spécification). Cet ensemble de concepts établit un langage adapté à l'écriture des spécifications sous ce point de vue, chacune d'entre elles formant un modèle du système exprimé dans les termes de ces concepts. La terminologie de chaque *langage de point de vue* et ses règles d'emploi sont définis par usage des concepts de modélisation par objets que définit la Rec. UIT-T X.902 | ISO/CEI 10746-2. La puissance d'expression de chacun des langages est suffisante pour spécifier une *fonction ODP*, une application ou une politique sous le point de vue correspondant. Tout système dont les spécifications sont conformes à ces langages est, tout au moins d'un point de vue architectural, un système ODP.

Dans le modèle de référence ODP, les différents langages de points de vue diffèrent par la force des contraintes que leur usage implique. Les langages des points de vue traitement et ingénierie, qui concernent l'organisation de la répartition et les solutions communes aux problèmes qu'elle pose, instaurent un grand nombre de contraintes qu'il est nécessaire d'observer. Ils garantissent de ce fait l'interfonctionnement et la portabilité des composants. Par contre, du fait du caractère général du modèle de référence ODP, il n'est possible de poser que peu de règles dans les langages d'entreprise et d'information. Ceux-ci se limitent donc à un jeu de concepts fondamentaux et à des directives relatives au domaine d'application de la modélisation des niveaux entreprise et information dans les systèmes répartis.

Il sera défini des contraintes plus sévères pour les langages d'entreprise et d'information lorsque ceux-ci serviront à spécifier des systèmes relevant de domaines d'application particuliers et d'entreprises spéciales. Dans le cas par exemple d'un système de courtage, les spécifications d'entreprise et d'information subissent les contraintes imposées par la norme de courtier.

### 8.1.2 Transparences à la répartition

Lorsque l'on conçoit un système réparti, il survient un certain nombre de difficultés qui résultent directement de la répartition: les composants du système sont hétérogènes, ils peuvent tomber isolément en panne, ils se trouvent à des endroits différents sinon même variables, et ainsi de suite. Ou bien on résout directement ces difficultés dans le cadre de la conception des applications, ou bien on choisit des solutions normalisées fondées sur les meilleurs usages.

S'il fait le choix de mécanismes normalisés, le concepteur d'application travaille dans un monde "transparent" à cette difficulté particulière, car elle n'apparaît plus. Il est dit du mécanisme normalisé qu'il offre une *transparence à la répartition*. Les concepteurs d'applications n'ont plus qu'à choisir celles des transparences à la répartition dont ils désirent admettre l'existence et à décider de l'endroit où, dans le système, elles doivent intervenir.

L'approche par transparences à la répartition peut mener directement au réemploi du logiciel. Faire le choix de transparences à la répartition lors de la conception du système peut permettre d'incorporer automatiquement aux outils de construction du système, compilateurs, éditeurs de liens ou gestionnaires de configuration, des réalisations bien connues des solutions normalisées. Le concepteur exprime les exigences du système sous la forme d'un énoncé simplifié du système demandé accompagné d'une déclaration des propriétés de transparence à la répartition dont il doit jouir.

Les transparences à la répartition que définit le modèle de référence ODP sont:

- a) la *transparence d'accès*, qui, pour permettre l'interfonctionnement des objets, masque les différences dans la représentation des *données* et dans les mécanismes d'*invocation*. Cette transparence à la répartition résout un grand nombre des problèmes d'interfonctionnement entre systèmes hétérogènes et sera donc généralement fournie par défaut;
- b) la *transparence aux défaillances*, qui, afin de garantir la résistance aux pannes, masque à un objet la défaillance éventuellement suivie de *reprise* des autres objets (ou de lui-même). Lorsque cette transparence est fournie, le concepteur peut travailler dans un monde idéal, débarrassé des défaillances de la classe correspondante;
- c) la *transparence à la localisation*, qui masque l'emploi d'informations de *localisation dans l'espace* lors de l'identification d'interfaces et d'établissement de liaisons avec elles. Cette transparence à la répartition donne de la désignation une vue logique indépendante de la localisation physique réelle;
- d) la *transparence à la migration*, qui masque à un objet la capacité que possède un système d'en changer la localisation. La migration sert souvent à équilibrer les charges et à réduire les délais;
- e) la *transparence à la relocalisation*, qui masque à des interfaces la relocalisation d'une interface à laquelle elles sont liées. La relocalisation permet à un système de continuer à fonctionner en dépit des incohérences temporaires que la migration ou le remplacement d'un objet quelconque peut amener dans la vue qu'en ont ses utilisateurs;
- f) la *transparence à la duplication*, qui masque la mise en œuvre d'une interface par un groupe d'objets faisant preuve de compatibilité de comportement. La duplication sert souvent à améliorer les performances et à accroître la disponibilité;
- g) la *transparence à la persistance*, qui masque à un objet la *désactivation* et la *réactivation* d'autres objets (ou de lui-même). La désactivation et la réactivation servent souvent à maintenir la *persistance* d'un objet lorsque le système est incapable de lui assurer de manière continue les fonctions de traitement, de stockage et de communication;
- h) la *transparence aux transactions*, qui masque la coordination permettant d'assurer la cohérence des activités au sein d'une configuration d'objets.

Dans toute spécification de système, la définition d'une transparence implique à la fois un jeu d'exigences et la transparence qui y satisfait. Le jeu d'exigences précise en quels points est requise la transparence, c'est-à-dire quelles interactions elle affecte. Il peut s'agir simplement d'une déclaration qui s'applique au système tout entier ou d'une déclaration plus spécifique à des interfaces particulières définissant, par exemple, les interactions constitutives d'une transaction ou choisissant les objets et les interfaces à prendre en charge par duplication. La solution prend la forme d'un jeu de règles permettant de transformer la spécification de la transparence à la répartition exigée en une spécification dans laquelle des interactions ou des objets déterminés sont complétés par incorporation des mécanismes qui produisent la transparence en question.

## 8.2 Langage d'entreprise

Le langage d'entreprise propose les concepts fondamentaux qui sont nécessaires pour représenter un système ODP dans le contexte de l'entreprise dans laquelle il fonctionne. Le but d'une spécification d'entreprise est d'exprimer les objectifs et les contraintes d'ordre politique qui pèsent sur le système considéré. A cette fin, le système est représenté par un ou plusieurs objets d'entreprise au sein d'une *communauté* d'objets d'entreprise qui représente cette entreprise et par les rôles auxquels participent ces objets. Ces rôles représentent, par exemple, les utilisateurs, les propriétaires ou les fournisseurs de l'information que traite le système. En créant un point de vue séparé pour transmettre cette information, on découple la spécification des objectifs assignés à un système de la manière dont ce système doit être réalisé.

Une des idées principales qui apparaît dans le langage d'entreprise est celle d'un contrat qui lie les acteurs des différents rôles au sein d'une communauté et exprime leurs obligations mutuelles. Un contrat peut exprimer les buts communs et les responsabilités qui particularisent les rôles dans une communauté, telle qu'une entreprise et ses clients ou qu'une administration et ses assujettis, comme s'ils étaient associés d'une manière particulière au sein d'une même entreprise.

Là où la nécessité en apparaîtra, la spécification d'entreprise exprimera aussi des aspects relatifs à la propriété des ressources et aux responsabilités portant sur le paiement des marchandises et des services afin de déterminer, par exemple, les contraintes qui pèsent sur les opérations comptables et les mécanismes de sécurité au sein de l'infrastructure qui sous-tend le système.

Il existe un type particulier de communauté, la fédération, qui est l'association d'un certain nombre de groupes qui, ressortissant à des autorités différentes et pouvant être représentés sous forme de domaines distincts, cherchent à coopérer pour atteindre un objectif commun. Comme l'évolution des systèmes répartis entraînera, pour partager les informations ou prendre en charge les intérêts commerciaux, une succession de fusions de sous-systèmes existants distincts et séparément gérés, la spécification de la *création* de fédérations et l'expression des règles applicables à ces structures constituent une part importante de la spécification des systèmes dans le point de vue entreprise.

Les domaines considérés dans une fédération sont soit administratifs, chacun d'eux pouvant, par exemple, répondre à des règles particulières de sécurité ou de gestion, soit techniques, chacun d'eux acceptant, par exemple, un même choix de logiciels ou de matériels. La spécification d'une fédération comporte celle des objectifs en matière d'interfonctionnement entre les différents domaines et des politiques qui régissent ces interfonctionnements.

Une fédération de domaines administratifs concerne l'interfonctionnement entre domaines appartenant à une même entreprise ou à différentes entreprises en vue du partage, de l'intégration et de la répartition des ressources entre les systèmes et les lieux en réponse aux besoins des utilisateurs. Une fédération de domaines techniques concerne l'intégration de systèmes qui diffèrent par leurs architectures, leurs ressources ou leurs performances; elle apporte la modularité qui autorise une croissance par étapes sans impact sur les applications en service. Les deux types de fédération coïncident souvent car des différences au plan administratif peuvent conduire à des différences quant aux choix de solutions techniques.

Entre domaines administratifs, l'une ou l'autre des administrations, ou les deux, peut vouloir imposer ses propres contrôles d'accès pour des raisons de sécurité, de gestion comptable et de surveillance, en plus des contrôles qu'imposent les objets eux-mêmes. C'est aux frontières administratives que s'effectue aussi le transfert des responsabilités de gestion, celles, par exemple, de l'allocation des ressources et de la garantie de sûreté de fonctionnement.

Parmi les politiques qui régissent le fonctionnement des fédérations figurent celles qui régissent les questions d'interfonctionnement. La spécification d'une fédération peut ainsi se rapporter à la spécification d'une fonction d'*intercepteur* dans la description d'ingénierie; les objectifs et les contraintes portant sur la fédération déterminent les contraintes qui pèsent sur la fourniture des services d'interception.

Une spécification d'entreprise définit les politiques qui régissent le comportement des communautés qu'elle spécifie. Ces politiques déterminent les actions exercées par les objets d'entreprise qui participent à ces communautés. Elles s'intéressent aux questions touchant à l'assignation et à l'accomplissement des obligations comme, par exemple, demande de livraison ou livraison, et à l'autorisation ou à l'interdiction des actions comme, par exemple, permission ou déni d'accès aux ressources du système. Les politiques peuvent concerner:

- a) La structuration de la communauté en termes de rôles et l'affectation des rôles aux objets d'entreprise. Des **règles communautaires**, par exemple, peuvent énoncer:
  - comment sont assignés les rôles et les responsabilités des objets d'entreprise au sein de la communauté;
  - comment les objets d'entreprise se relient à la structure communautaire (par exemple, hiérarchie ou égalitarisme).

Une spécification d'entreprise peut aussi comporter des **règles d'entreprise** qui expriment:

- l'entreprise en tant qu'entité faisant des affaires;
  - les exigences portant sur la gestion comptable;
  - l'évolution de l'entreprise pour accomplir ses objectifs.
- b) Les interactions autorisées entre objets d'entreprise tenant différents rôles, c'est-à-dire le contrôle d'accès. Des **règles de sécurité**, par exemple, peuvent définir:
    - les relations entre rôle, *activité et* objet, ainsi que les exigences qu'elles portent sur les activités et les objets en matière d'intégrité et de confidentialité;
    - les règles applicables à la détection des menaces pesant sur la sécurité;
    - les règles applicables à la protection contre les menaces pesant sur la sécurité;
    - les règles applicables à la limitation des dommages causés par toute atteinte à la sécurité.

- c) La responsabilité déléguée aux objets de l'entreprise. Des **règles de délimitation d'autorité**, par exemple, servent à affecter:
- des privilèges aux objets d'entreprise (confiance);
  - l'autorisation donnée ou l'interdiction faite aux objets d'entreprise de mener des actions.
- d) La comptabilisation de l'usage des ressources. Des **règles d'usage des ressources**, par exemple, définissent les contraintes que peuvent imposer de l'extérieur:
- des organismes de réglementation;
  - les demandes du marché;
  - l'environnement;
- selon que l'usage fait de la ressource est:
- public;
  - privé;
  - de tierce partie.
- e) La propriété des ressources. Des **règles de transfert**, par exemple, peuvent formuler les changements de propriétaire ou de responsable des ressources parmi les objets d'entreprise.
- f) Le sociétariat des fédérations. La spécification d'entreprise peut par exemple inclure des **règles de domaine** qui spécifient:
- les règles applicables au sociétariat d'un domaine;
  - les règles applicables aux interactions entre domaines du même type;
  - les règles applicables à la désignation dans un domaine.

Il est prévu qu'il existera différentes notations pour exprimer les spécifications d'entreprise dans le cas de structures d'organisation et d'usages commerciaux particuliers. Le modèle de référence ODP exige que soit établie une spécification adéquate, mais impose peu de contraintes sur la forme que doit prendre l'organisation.

L'évaluation de la conformité à la spécification d'entreprise d'un système demande de lier les exigences exprimées dans la spécification, qui sont du genre temps de réponse pour accomplir une obligation donnée, à des ensembles d'observations du comportement du système effectuées en des *points de conformité* précisés dans les spécifications d'ingénierie et de technologie et de mesurer le degré de cohérence entre les exigences et les observations.

### 8.3 Langage d'information

Pour que le système réparti se comporte comme prévu, tous ses composants doivent partager une même compréhension de l'information qu'ils échangent au cours de leurs communications. Certains de ces éléments d'information sont manipulés d'une manière ou d'une autre par un grand nombre des objets du système. Afin d'assurer une interprétation cohérente de ces éléments d'information, le langage d'information définit des concepts qui servent à spécifier la signification de l'information que conserve et manipule un système ODP, indépendamment de la manière dont seront réalisées les fonctions de traitement de l'information elles-mêmes.

Les informations que contient un système ODP sur des entités du monde réel, dont le système ODP lui-même, sont représentées dans une spécification d'information en termes d'objets d'information, de leurs relations et de leur comportement. Les éléments d'information fondamentaux sont représentés par des objets d'information atomiques. Les informations plus compliquées sont représentées par des objets d'information composites qui, chacun, expriment des relations portant sur un ensemble d'objets d'information constitutifs.

Exactement comme dans le cas familier de la modélisation des données, la spécification d'information se compose d'un ensemble de schémas apparentés, à savoir les schémas *d'invariant*, *statique* et *dynamique*.

Un *schéma d'invariant* exprime des relations entre objets d'information qui doivent rester toujours vraies dans tous les comportements valides du système. C'est ainsi qu'un schéma d'invariant pour un compte bancaire pourrait spécifier que le solde doit toujours être positif ou nul, la banque n'autorisant pas de découvert.

Un *schéma statique* exprime des affirmations qui doivent être vraies à un instant donné. Les schémas statiques sont communément utilisés pour spécifier l'état initial d'un objet d'information. L'état initial d'un objet compte bancaire, par exemple, se compose d'un solde de 0 franc et du retrait journalier qui est aussi de 0 franc. Un autre schéma statique pourrait exprimer que le retrait journalier est tous les jours de 0 franc à 0 heure (il est à noter que ce schéma ne fait aucune restriction sur le solde à ce moment).

Un *schéma dynamique* spécifie comment l'information peut évoluer lorsque le système fonctionne. Un compte bancaire, par exemple, demanderait un schéma dynamique pour représenter le dépôt d'argent, le retrait d'argent, le paiement des intérêts et les frais afférents à un compte crédit. Un schéma dynamique pourrait n'être applicable que dans certaines circonstances qui pourraient être spécifiées par le moyen d'un schéma statique. Le schéma dynamique pour le retrait de N francs, par exemple, pourrait spécifier que le solde du compte est diminué de N francs à condition que le total des retraits effectués ce jour-là ne dépasse pas 2500 francs. Aucun schéma dynamique ne peut spécifier d'état résultant qui violerait la contrainte invariante selon laquelle il ne peut être retiré d'argent que s'il est disponible au compte.

En plus de la description des transitions d'état, les schémas dynamiques sont aussi capables de créer ou de supprimer des objets composants. Par ce moyen, il est possible de modéliser la spécification d'information d'un système ODP toute entière sous la forme d'un seul objet d'information, *a priori* composite.

Ces schémas s'appliquent au système dans sa totalité ou à des domaines particuliers qui lui appartiennent. En particulier, s'agissant de systèmes étendus et rapidement variables, la réconciliation et la fédération de domaines d'information distincts formeront l'une des tâches les plus importantes de la gestion de l'information.

Il n'est pas nécessaire que les schémas relatifs aux objets composites fassent référence à tous les composants de l'objet d'information. Ces schémas peuvent se composer des schémas de leurs objets constitutifs, pourvu qu'une telle composition ait un sens. L'encapsulation des objets d'information se rapporte au niveau d'abstraction de la description en cause. Ainsi, les schémas applicables à des objets d'information peuvent à un certain niveau d'abstraction faire référence aux caractères internes de leurs objets constitutifs, bien que ceci puisse ne pas être possible à un niveau d'abstraction plus élevé. Il est ainsi possible de spécifier des phrases aussi compliquées que "le numéro de téléphone du client des comptes duquel plus de 2000 francs ont été retirés aujourd'hui".

Quelques-uns des éléments visibles au niveau entreprise le restent au niveau information et réciproquement. Par exemple, une activité vue du point de vue entreprise peut apparaître dans le point de vue information sous forme de spécification d'un traitement qui ferait subir une transition d'état à une entité d'information.

Des notations différentes pour la spécification d'information permettent de modéliser de différentes manières les propriétés de l'information. On peut faire porter l'accent sur la classification et la reclassification des types d'information ou sur les états et le comportement des objets d'information. Dans certains langages de spécification, les objets d'information atomiques sont représentés sous forme de valeurs. L'approche à suivre dépendra de la technique de modélisation et de la notation en usage.

L'évaluation de la conformité à la spécification d'information d'un système demande de lier les exigences exprimées dans la spécification, par exemple dans un schéma d'invariant, à des ensembles d'observations du comportement du système effectuées en des points de conformité précisés dans les spécifications d'ingénierie et de technologie et de mesurer le degré de cohérence entre les exigences et les observations.

## 8.4 Langage de traitement

Le point de vue traitement s'intéresse directement à la répartition du traitement, mais pas aux mécanismes d'interaction grâce auxquels la répartition devient possible. La spécification de traitement décompose le système en objets qui remplissent des fonctions individuelles et qui interagissent mutuellement sur des interfaces bien définies. Elle fournit ainsi le fondement des décisions sur la manière de répartir les travaux à effectuer puisqu'il est possible de placer les interfaces en toute indépendance, supposition faite qu'il soit possible de définir dans la spécification d'ingénierie les mécanismes de communication capables de prendre en charge le comportement à ces interfaces.

Le cœur du langage de traitement est le modèle par objets qui définit:

- la forme d'interface que peut présenter un objet;
- la manière dont les interfaces peuvent être liées et les formes d'interactions qui peuvent s'y dérouler;
- les actions qu'un objet peut exécuter dont, en particulier, la création de nouveaux objets et de nouvelles interfaces, ainsi que l'établissement de liaisons.

Le modèle de traitement fournit la base qui permet d'assurer que les langages de spécification, les langages de programmation et les mécanismes de communication opéreront tous de manière cohérente, condition d'interfonctionnement ouvert et de portabilité des composants.

Le langage de traitement donne au concepteur les moyens d'exprimer, en termes de contrats d'environnement associés aux interfaces particulières et de liaisons d'interfaces, les contraintes qui pèsent sur la répartition d'une application, sans qu'il ait à spécifier le niveau de répartition effectif. De là provient la garantie que les applications ne font aucune hypothèse non explicitée quant à la répartition de leurs composants. De ce fait, la configuration et le niveau de répartition des matériels sur lesquels tournent les applications ODP sont aisément modifiables, dans le cadre des contraintes d'environnement énoncées, sans impact grave sur le logiciel d'application.

Le langage de traitement n'interdit pas d'employer dans des environnements répartis des logiciels conçus pour des environnements centralisés. Il permet d'enchaîner les applications actuelles sous forme de composants non répartis d'applications réparties plus importantes. C'est la source d'une approche à la prestation de la répartition dont le caractère évolutif protège les investissements réalisés sur les logiciels en service.

Les interactions qui se produisent entre objets de traitement sont essentiellement asynchrones; elles peuvent prendre trois formes:

- les *opérations*, qui ressemblent aux procédures, et sont appelées sur des interfaces désignées;
- les *flux*, qui sont des abstractions de successions continues de données entre interfaces;
- les *signaux*, qui sont des interactions atomiques élémentaires.

Les opérations sont le reflet du paradigme client/serveur. Une opération est une interaction qui se produit entre un *objet client* et un *objet serveur* qui appelle (ou invoque) l'exécution d'une quelconque fonction par le serveur. Il y a deux types d'opérations:

- l'*interrogation*, dans laquelle le serveur renvoie une réponse (ou *terminaison*) à la demande du client;
- l'*annonce*, dans laquelle il n'y a pas de réponse à la demande du client.

La notion de terminaison est une généralisation des concepts de résultat et d'exception que l'on trouve dans de nombreux langages de programmation, qu'ils fassent appel aux objets ou non.

L'exécution des opérations se produit dans l'espace et dans le temps. Il en résulte qu'en cas d'échec d'une opération, cette défaillance ne touche pas nécessairement tous les participants, qui peuvent, de plus, l'observer à des moments différents. La capacité du client à observer les défaillances et à réagir en conséquence n'est pas la même dans le cas d'une interrogation ou dans celui d'une annonce.

S'il s'agit d'une interrogation, le mécanisme d'acquiescement bidirectionnel garantit à la fois que le client reçoit confirmation de l'exécution de la fonction et que, si le *fil d'exécution* de l'activité du client appelle une *chaîne* d'interrogations, le serveur répond dans l'ordre d'émission par le client.

S'il s'agit d'une annonce, toute garantie sur l'exécution des demandes et sur leur ordre d'exécution ne peut provenir que des contrats d'environnement qui s'appliquent aux opérations.

Les flux peuvent servir à modéliser, par exemple, le débit d'information sonore ou visuelle d'une application multimédia ou de services de télécommunication vocaux, ou le déroulement continu des lectures périodiques de capteurs dans des applications de commande de processus. Un flux se caractérise par son nom et par son type, qui spécifie la nature et le format des données échangées. Le modèle de traitement ne spécifie pas la sémantique exacte des flux. Il peut en effet exister, selon le domaine d'application, de nombreuses sémantiques de flux.

Les signaux constituent le niveau le plus bas de description des interactions entre objets de traitement. Un signal est un couple atomique d'actions partagées qui résulte en une communication à sens unique d'un objet de traitement *initiateur* vers un objet de traitement *répondeur* (dans ce contexte, "répondeur" veut dire "qui accepte" la communication). En conséquence:

- le signal se produit à un moment bien défini qui constitue donc un point de référence pour les mesures, comme, par exemple, pour les observations de qualité de service;
- une défaillance est identique pour tous les participants, qui tous la voient.

Dans de nombreux cas, un signal correspondra, en termes de réalisation, à un événement observable en quelque point physique, mais la définition du concept n'interdit pas de mettre en œuvre des signaux au moyen de mécanismes transactionnels qui fournissent les garanties de comportement requises.

Une opération ou un flux peuvent s'analyser en tant que combinaison de plusieurs signaux. On peut par exemple interpréter une interrogation comme la séquence des signaux émission de demande par l'objet client, réception de demande par l'objet serveur, émission de terminaison par le serveur, réception de terminaison par le client. Par contre, comme la sémantique exacte des flux n'est pas précisée par le modèle de traitement, leur mappage sur les signaux n'est pas défini. Il est nécessaire de modéliser les opérations et les flux en termes de signaux pour pouvoir définir les caractéristiques de qualité de service de bout en bout ainsi que les opérations de liaisons multiparties et les liaisons entre interfaces d'espèces différentes comme, par exemple, des liaisons entre interfaces flux et *interfaces opération*.

#### 8.4.1 Interfaces de traitement

Une interface de traitement se caractérise par une signature, un comportement et un contrat d'environnement.

La signature dépend du type d'interface, qui peut être opération, flux ou signal:

- Une *interface opération* a une signature qui définit quel jeu d'opérations cette interface met en œuvre et si l'interface joue le rôle de client ou de serveur pour ce jeu d'opérations.
- Une *interface flux* a une signature qui définit quel jeu de flux cette interface met en œuvre et, pour chaque flux, si l'interface joue le rôle de producteur ou de consommateur.
- Une *interface signal* a une signature qui définit quel jeu de signaux cette interface met en œuvre et, pour chaque signal, si l'interface joue le rôle d'initiateur ou de répondeur.

Le comportement est décrit par les séquences d'actions permises aux objets de traitement qui sont associés à l'interface. Le comportement peut comprendre des actions internes à l'objet. Il est soumis aux contraintes d'environnement de l'objet résultant, en particulier, d'interactions intervenant sur d'autres interfaces.

Il est important de noter que toute spécification d'interface comporte aussi un contrat d'environnement qui spécifie un jeu de contraintes de qualité de service portant sur un objet de traitement et son environnement. Si l'environnement, qui est constitué d'autres objets de traitement et de l'infrastructure sous-jacente, tient le niveau demandé de qualité de service, alors il est garanti par construction que l'objet lui-même fournira un certain niveau de qualité de service. Les caractéristiques de qualité de service spécifiées à une interface expriment à la fois les exigences qu'elle porte sur son environnement et la qualité de service qu'exhibe l'objet s'il est placé dans un environnement qui répond aux exigences. Le langage de traitement ne normalise pas de notation particulière pour la spécification de la qualité de service.

#### 8.4.2 Modèle de liaison

Les interactions entre des interfaces données de traitement ne peuvent intervenir que s'il a été établi une liaison, c'est-à-dire un chemin de communication quelconque, entre ces interfaces. Le langage de traitement spécifie des actions de *liaison explicite* dans les deux cas des interfaces opération et flux. Il spécifie en plus que dans le cas des interfaces opération la liaison peut être *implicite*, afin de pouvoir utiliser des notations qui ne donnent pas le moyen d'exprimer les actions de mise en liaison. Une *liaison implicite* ne peut intervenir que dans le cas d'interfaces opération car, dans les autres cas, il n'est pas évident de savoir où se prend l'initiative dans la liaison par référence aux interactions ultérieures.

Quand la liaison est implicite, une invocation par l'objet client entraîne la formation d'une liaison entre une interface adéquate du client et l'interface du serveur pour l'exécution de l'opération d'interrogation ou d'annonce. Le langage de traitement ne décide pas si l'interface client est ou n'est pas supprimée à la fin du processus. Il est à noter que la liaison implicite ne donne pas le moyen de faire référence à un contrat d'environnement pour cette liaison.

Quand la liaison est explicite, elle est définie par référence à deux sortes d'actions de liaison, les *actions de liaison primitives* et les *actions de liaison composites*. Ces actions de liaison ne s'appliquent qu'en cas de liaison explicite.

Une action de liaison primitive permet de lier deux interfaces appartenant à un même objet de traitement ou à deux objets de traitement différents. Les interfaces doivent appartenir au même type, qui peut être opération, flux ou signal. Une action de liaison primitive est exécutée par l'un des objets concernés. Elle a pour effet de doter chaque interface de l'information requise pour que l'interaction puisse avoir lieu, c'est-à-dire l'identité de l'autre interface concernée. Il n'apparaît pas nécessaire de définir une action de *fin de liaison* explicite, mais, évidemment, la suppression de l'une des interfaces supprime aussi la liaison. Une action de liaison primitive exige que les interfaces concernées appartiennent au même type et soient munies de types de signatures et de rôles complémentaires (par exemple que l'une soit client et l'autre serveur).

Une action de liaison composite permet de lier deux interfaces, ou plus, du même type ou de types différents, par le truchement d'un *objet de liaison* (voir la Figure 2).

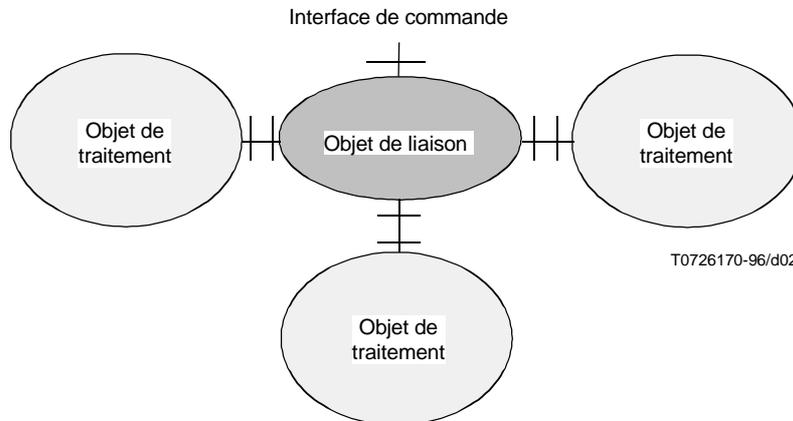


Figure 2 – Liaison composite

L'action peut être exercée par l'un des objets de traitement impliqués dans la liaison ou par un objet de traitement n'appartenant pas à la liaison. Elle a pour effet de créer une instance d'objet de traitement, l'objet de liaison, qui rend possible la liaison. L'objet de liaison crée un jeu d'interfaces adéquat et se sert d'actions de liaison primitives pour établir des liaisons avec les interfaces à lier. Il crée aussi des instances d'interfaces de commande au travers desquelles pourront être gérées ses opérations et il envoie à l'objet de traitement initiateur les identifiants d'interfaces correspondant à toutes ces interfaces.

Le comportement des objets de liaison reflète la sémantique de la communication qu'ils mettent en œuvre. Le modèle de référence ne restreint pas les types d'objets de liaison, par référence au fait qu'il existe un grand nombre de structures de communication possibles entre objets. Il est néanmoins possible de normaliser des classes utiles d'objets de liaison par relation avec les classes d'applications. Certains objets de liaison sont capables de spécifier en particulier le fonctionnement de liaisons entre plusieurs partenaires ou celui de liaisons complexes entre, par exemple, différentes interfaces opération ou flux et entre des interfaces opération et des interfaces flux.

Comme pour les autres objets, il est possible de qualifier les objets de liaison par des déclarations de qualité de service (bornes mises sur les délais de transmission de bout en bout ou sur la gigue du délai de bout en bout visibles à une interface de réception) qui en restreignent encore davantage le bon comportement. Lorsque sont exprimées des déclarations de cette sorte, les interfaces impliquées dans les liaisons primitives d'un objet de liaison doivent être des interfaces signal, car l'atomicité de leur nature rend possible la spécification dans le temps et dans l'espace des points où pourront s'exercer les observations de qualité de service.

Les interfaces de commande d'un objet de liaison permettent la *suppression* de la liaison. Elles permettent aussi d'en commander les opérations et le niveau de qualité de service offert. Des exemples des capacités qui pourraient être proposées sont:

- a) une commande de *notification* des *erreurs* qui empêchent le fonctionnement de l'objet de liaison: cette capacité permettrait de spécifier une interface sur laquelle l'objet pourrait lancer une opération de notification si une défaillance venait à causer une rupture de la liaison;
- b) une commande de liaison dynamique multiparties, donnant les moyens d'ajouter de nouveaux consommateurs et de retirer des consommateurs présents;
- c) une invocation de groupe, rendant disponibles dans le langage de traitement des invocations atomiques multipartenaires et autorisant l'adjonction de membres au groupe et leur retrait du groupe;

- d) une gestion de la qualité de service associée à la liaison, rendant possible la manipulation de certaines caractéristiques de qualité de service. Cette forme de gestion aurait un intérêt particulier dans le cas des liaisons flux relatives à des applications multimédias;
- e) une notification des événements d'intérêt pour l'application. Un événement pourrait par exemple être signalé au commencement ou à la fin d'une période de silence dans un flux sonore.

Dans le cas de liaison entre interfaces flux, il est permis à la liaison de s'abstraire des règles de composition de flux spécifiques à l'application. Dans le cas le plus simple, la liaison représente un flux unique circulant d'une interface de production à une interface de consommation, par exemple d'un système d'archivage audio à un haut-parleur. Mais les règles de composition pourraient devenir plus complexes:

- a) il est possible de créer et de gérer un chemin duplex sous la forme d'une liaison unique. Le chemin duplex relie les aspects producteur de l'interface appartenant à chaque objet de traitement aux aspects consommateur de l'interface appartenant à l'autre;
- b) il est possible de relier un certain nombre d'interfaces duplex par un objet de liaison qui cache les règles d'un système de téléconférence afin de permettre de délivrer le flux issu d'un *producteur* particulier (le locuteur qui est en train de parler) à tous les consommateurs. L'application pourrait exercer divers niveaux de commande par le truchement de l'interface de commande de l'objet liaison afin d'exercer un contrôle de flux explicite;
- c) il est possible de combiner les flux issus d'un certain nombre de producteurs pour apporter à un consommateur unique un flux composite. Par exemple, on pourrait combiner un flux vidéo en provenance d'une source avec un flux audio en provenance d'une autre source pour construire un flux télévisuel unique comportant l'image et le commentaire associé. Dans ce cas, l'interface de commande pourrait autoriser la manipulation des mécanismes de synchronisation des flux d'ingénierie pour participer à la synchronisation du son avec les mouvements des lèvres.

Une interface est susceptible de liaisons multiples. En cas de liaison implicite, il faut que chacune des liaisons de l'interface soit identifiée par l'interface de serveur impliquée. En cas de liaison explicite, il faut que chacune des liaisons soit identifiée par l'objet de liaison impliqué.

### 8.4.3 Typage et sous-typage dans les interfaces de traitement

Dans le langage de traitement, les interfaces sont fortement typées pour que soit poussée au maximum la possibilité de vérifier au plus tôt la cohérence des programmes répartis conformes à ce langage. Les types d'interfaces sont liés par une relation de sous-typage qui définit les conditions minimales à imposer pour que les interactions entre objets aient un sens.

Le type de signature d'une interface définit la forme et la sorte des interactions disponibles à cette interface; le sous-typage d'une signature spécifie les exigences minimales pour qu'une interface puisse se substituer à une autre. Les règles sont fondées sur la sémantique d'interaction des interfaces de traitement. Elles suffisent à assurer qu'une interface de substitution soit en mesure d'interpréter correctement la structure de n'importe quelle interaction lorsqu'elle survient. Il faut naturellement aussi que les interfaces concordent sur le plan de la sémantique des données transférées, mais il n'est pas possible de définir de règles générales pour arriver à cette concordance.

Pour chaque signal permis à une interface, la *signature d'interface signal* définit son nom, ses paramètres et le rôle, *initiateur* ou répondeur, que joue l'interface à son égard. Pour chaque sorte d'opération permise à une interface, la *signature d'interface opération* définit son nom, le nombre et le type de ses arguments, ainsi que, dans le cas des interrogations, l'ensemble des conclusions, ou terminaisons, qui sont possibles pour l'opération. Le nom de chaque terminaison, le nombre et le type de ses arguments sont définis.

Les règles de sous-typage applicables aux signatures d'interfaces signal et opération sont définies dans l'Annexe A à la Rec. UIT-T X.903 | ISO/CEI 10746-3. Il est à noter que les sémantiques d'interactions concernant d'autres types d'interfaces que les interfaces opération et flux sont exprimables en termes de signaux puisque ceux-ci forment les briques élémentaires à partir desquelles est modélisable n'importe quel type d'interaction de niveau plus élevé.

Le type d'une interface flux s'exprime sous forme d'un ensemble de flux constitutifs possédant chacun un type de base que les mécanismes sous-jacents disponibles sont capables de prendre en charge. Des flux singuliers sonores ou visuels sont des exemples de tels types de base. Chaque flux constitutif s'écoule dans un seul sens, entrant ou sortant de la liaison. Tout comme l'organisation des arguments forme une signature dans une interface opération, la description de type d'interface organise les flux constitutifs pour former une signature de flux. Une interface flux peut consister en un certain nombre de flux apparentés s'écoulant soit dans une même direction, soit dans des directions opposées. Mais, dans le cas des flux, le type d'interface lui-même peut servir à représenter un flux plus complexe, en sorte que, si les types sont établis en une succession d'étapes, les flux peuvent s'organiser en une hiérarchie à niveaux multiples.

Exemples d'interfaces flux:

- a) un flux audio unique issu d'une source de sons;
- b) un flux audio unique entrant dans un puits de sons;
- c) un type de discours duplex comportant un flux entrant et un flux sortant pour représenter la vue qu'a l'utilisateur des aspects vocaux d'un service téléphonique;
- d) un signal télévisuel complexe comprenant à la fois des composantes audio et vidéo;
- e) un type plus compliqué correspondant à une application demandant la combinaison de plusieurs flux audiovisuels pour représenter une réalité virtuelle.

Du fait que chacun des flux d'une interface flux est nanti d'une direction, les types d'interfaces devront en général apparaître par couples caractérisés par l'inversion de tous les flux rencontrés. Si toutefois le type d'interface propose un ensemble équivalent de flux entrants et sortants, les deux types appariés deviennent eux-mêmes équivalents. Pour dénoter cette situation, la notation de définition d'interface pourra soit permettre une expression abrégée, soit permettre une description simultanée des deux formes apparentées.

A tout système de types d'interfaces flux sera associé un jeu de règles de sous-typage. Elles diffèrent des règles de sous-typage des interfaces opération en ce qu'elles sont conçues pour permettre de communiquer entre des objets de traitement dont les interfaces flux ont des capacités différentes. Une interface audio peut par exemple être considérée comme sous-type d'une interface composée audiovisuelle, ce qui permet à un client du téléphone de communiquer avec un client d'un système de visiophonie. La forme de sous-typage la plus appropriée dépendra de l'application, en sorte que le choix de la variante adéquate de sous-typage forme fait partie du processus de conception.

D'une manière générale, les règles de sous-typage de flux se traitent en deux étapes:

- a) identification des correspondances entre flux primitifs des deux types et décision sur la suffisance des correspondances ainsi trouvées pour qu'existe une relation de sous-typage;
- b) comparaison, aspects de qualité de service y compris, des types de chacun des flux primitifs, pour déterminer s'il existe une relation de sous-typage.

Il n'est pas possible de définir complètement des règles de sous-typage pour les interfaces flux car elles dépendent du détail des interactions dont les définitions des flux concernés donnent l'abstraction.

#### **8.4.4 Portabilité**

Le langage de traitement définit les actions qu'un objet peut exercer et énumère les modes de défaillance dont ces actions sont susceptibles. Il définit donc un modèle de programmation par objets relatif à une machine virtuelle générale dont la réalisation résulte de l'application des règles d'ingénierie et de technologie.

Il est possible de définir différents ensembles de règles de portabilité, chacun d'entre eux spécifiant un sous-ensemble particulier des actions définies par le modèle de traitement. Un ensemble de règles de portabilité détermine les exigences que la prise en compte de la portabilité des objets entre des environnements capables de traiter ces exigences fait porter sur la notation de traitement. Le modèle de référence ODP lui-même définit, selon l'ensemble des actions prises en charge, un environnement de portabilité de base et un environnement de portabilité complet.

## 8.5 Langage d'ingénierie

Le langage d'ingénierie porte sur la manière dont sont réalisées les interactions entre objets et sur les ressources nécessaires à cet effet. Il définit les concepts qui servent à décrire l'infrastructure requise pour assurer aux interactions entre objets les diverses transparences à la répartition, ainsi que les règles qui servent à organiser des *canaux* de communication entre les objets et à organiser les systèmes aux fins de gestion des ressources.

Par conséquent, alors que le point de vue traitement se soucie du quand et du pourquoi des interactions entre objets, le point de vue ingénierie se soucie du comment. Dans le langage d'ingénierie, on s'intéresse principalement à la mise en œuvre des interactions entre objets de traitement. Il existe par conséquent des liens très étroits entre les descriptions dans ces points de vue: les objets de traitement sont visibles dans le point de vue ingénierie sous forme d'*objets d'ingénierie de base* et les liaisons de traitement, qu'elles soient implicites ou explicites, sont visibles sous forme soit de canaux, soit de liaisons locales.

Les concepts et les règles suffisent à la spécification des interfaces internes à l'infrastructure permettant de définir des points de conformité distincts pour les différentes transparences et offrant la possibilité de normaliser une infrastructure générique au sein de laquelle pourront être placés des modules normalisés de réalisation des transparences.

### 8.5.1 Grappes, capsules et nœuds

Le langage d'ingénierie traite des objets d'ingénierie de base et de certains autres objets d'ingénierie qui les prennent en charge. Il relie ces objets aux ressources disponibles dans le système par détermination d'une série de regroupements emboîtés.

A la couche la plus externe, les objets d'ingénierie se trouvent localisés physiquement auprès de ressources de traitement auxquelles ils sont associés par regroupement en *nœuds*. On peut imaginer ces nœuds comme la représentation de systèmes de traitement de l'information ressortissant de gestions indépendantes. Un nœud est toute entité dont les ressources sont si étroitement intégrées que le concepteur de système peut les traiter comme un tout. Ainsi, un système parallèle à couplage fort peut être considéré comme un nœud, pourvu qu'il obéisse à une seule politique d'ordonnement et d'allocation, sous un seul système d'exploitation.

Le nœud est sous le contrôle d'un *noyau* qui porte la responsabilité de la mise en route, de la création de groupes d'objets d'ingénierie, de la mise à disposition de moyens de communication et de la fourniture de services de base dont l'horloge et l'affectation d'identifiants uniques.

Il peut exister au sein d'un nœud un certain nombre de *capsules*. Une capsule possède des capacités de stockage et une part des ressources de traitement du nœud. On peut la voir comme un processus protégé traditionnel muni de son propre espace d'adressage. C'est donc l'unité de protection. Elle forme en général le plus petit module de défaillance que reconnaît le système d'exploitation. A chaque capsule est associé un objet d'ingénierie spécial, le *gestionnaire de capsule* qui, aux fins de description, commande la capsule par le biais d'interactions.

Une capsule contiendra normalement de nombreux objets d'ingénierie; le regroupement d'objets en capsules sert à réduire le coût des interactions entre objets. La communication entre processus traditionnels est en effet lente et coûteuse à cause des vérifications auxquelles il faut procéder; mais on peut faire confiance à la capacité des outils de compilation qui construisent les capsules à *valider* et à structurer les interactions entre des objets d'ingénierie si étroitement liés qu'il est possible de les laisser partager des ressources. Les ressources internes à une capsule seront sous le contrôle de quelque système d'exécution spécifique du langage.

Le plus petit regroupement d'objets d'ingénierie prend la forme d'un ensemble de grappes au sein d'une capsule. Les objets d'une capsule sont regroupés afin d'en réduire le coût de manipulation. La Figure 3 illustre grappes, capsules et nœuds. Il est possible de poser simultanément des *points de reprise* sur tous les objets d'ingénierie d'une grappe, de les transférer d'un bloc vers une mémoire permanente, de les réactiver, ou de les déplacer vers un autre nœud. Manipuler ainsi des grappes complètes par une opération unique ouvre la voie à une gestion économiquement acceptable de systèmes composés de très petits objets. Un système d'information géographique pourrait, par exemple, traiter chaque point d'une carte comme un objet d'ingénierie, mais sans être en mesure de supporter le coût résultant de l'attribution à chacun de ces objets d'une existence autonome. Il est possible de porter à un haut niveau l'optimisation de la communication entre objets d'ingénierie appartenant à une grappe puisque ces objets sont créés simultanément, dans le même langage et sont supposés rester ensemble.

Une interaction qui intervient au sein d'une grappe pourrait donc être prise en charge par une simple méthode d'invocation locale ou par toute autre solution équivalente.

Des interactions avec un objet *gestionnaire de grappe* servent à régir les grappes et à en déclencher les actions.

8.5.2 Canaux

Lorsque se produisent des interactions entre objets d'ingénierie appartenant à des grappes différentes, il faut disposer d'un bon nombre de mécanismes sous-jacents. Même si à ce moment les objets se trouvent dans la même capsule ou dans le même nœud, il faut disposer de mécanismes capables de faire face au cas où l'un de ces objets s'arrêterait, tomberait en panne ou s'en irait ailleurs. L'ensemble des mécanismes requis pour atteindre cet objectif constitue un canal, lui-même composé d'un ensemble d'objets d'ingénierie en interaction (voir la Figure 4).

Les objets d'ingénierie appartenant à un canal se répartissent en trois types selon la tâche qui leur est confiée. Les *talons* s'attachent à l'information transmise au cours d'une interaction, les *objets lieurs* sont chargés d'entretenir l'association parmi l'ensemble des objets d'ingénierie de base que relie le canal et les *objets de protocole* gèrent la communication effective.

Les talons sont directement en interaction avec les objets d'ingénierie de base dont ils assument la charge; ils exécutent des fonctions telles que l'ordonnancement et la dispersion des paramètres ou la journalisation des informations relatives à l'interaction en cours. Les talons ont donc besoin d'accéder à l'information sur le type de l'interaction ou, plus généralement, sur le type de l'interface prise en charge. Ils se distinguent par là des objets lieurs et des objets de protocole qui transportent des messages complets sans se soucier de leur structure interne.

Suivant la conception du système, un talon peut être directement associé à un objet d'ingénierie de base particulier ou être partagé entre un certain nombre de ces objets. Le partage implique généralement de transférer un complément d'information pour pouvoir nommer les objets pris en charge et donc pouvoir les distinguer entre eux.

Les objets lieurs portent la responsabilité de la résolution de bien des problèmes qui résultent de la répartition. Cet objet établit la liaison au moment de la création du canal et, par la suite, maintient l'intégrité du canal de bout en bout. Il doit donc traiter des changements de configuration et des défaillances des objets ou de la communication et faire le suivi des autres extrémités lorsque les objets se déplacent, tombent en panne ou sont remplacés (processus de relocalisation d'un objet). Les objets lieurs sont donc impliqués dans la fourniture de nombre de transparences à la répartition.

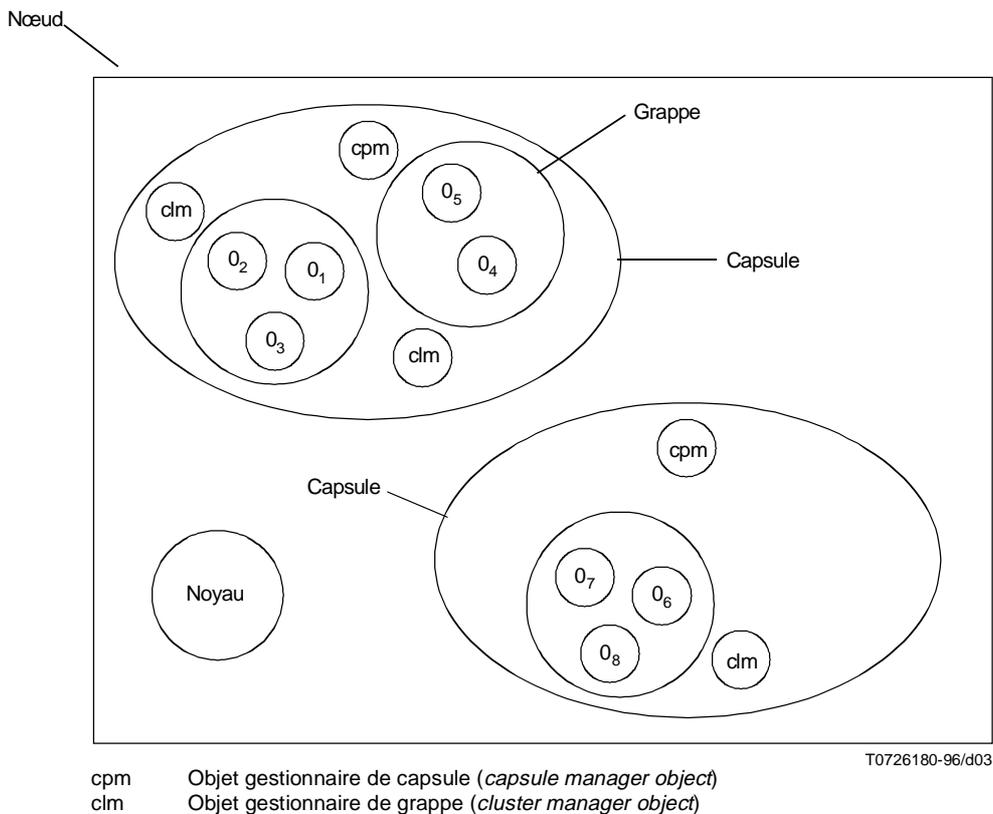
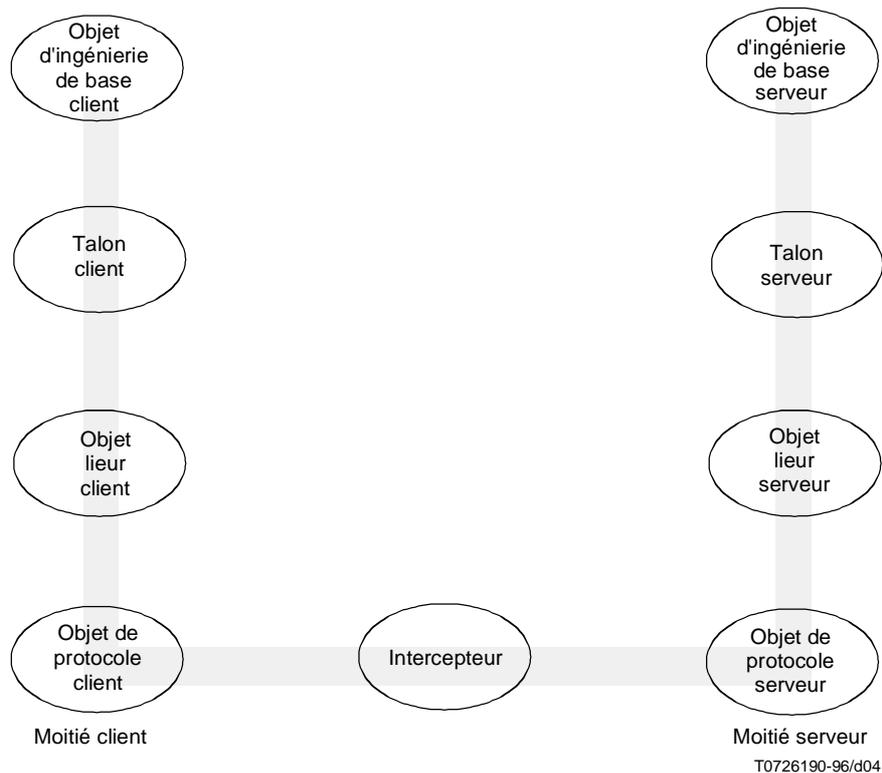


Figure 3 – Capsules, grappes et nœuds



**Figure 4 – Exemple de canal client-serveur**

Les objets de protocole sont chargés de réaliser entre les objets lieurs aux services desquels ils sont des communications de qualité et de fiabilité suffisantes. Non seulement ont-ils à manipuler tout protocole entre homologues en usage, mais ils doivent aussi donner accès, si nécessaire, à des services d'ordre général tels que des services d'annuaire pour l'obtention des adresses.

Il est possible qu'un quelconque objet de l'un de ces trois types ait à communiquer avec d'autres parties du système afin d'obtenir l'information dont il a besoin pour faire son travail ou pour fournir de l'*information de gestion* à d'autres objets d'ingénierie. Comme cette communication entre ces objets et des endroits quelconques du système peut à son tour avoir besoin de diverses transparences à la répartition, elle s'effectue aussi au moyen de canaux; de ce point de vue, les objets d'ingénierie situés au sein d'un canal jouent le rôle d'objets d'ingénierie de base dans un autre. De même, chacun de ces objets peut accepter des fournitures de commande à travers lesquelles il peut être géré. Ainsi par exemple il peut se faire qu'un objet de protocole fournisse une interface de commande au travers de laquelle il est possible d'ajuster l'objectif de qualité de service du canal.

Dans les cas où le canal doit traverser une frontière d'ordre technique ou organisationnel, il peut devenir nécessaire d'ajouter des vérifications ou des transformations afin d'observer les exigences imposées par l'un et l'autre côté. Ces fonctions sont remplies par des intercepteurs, décrits ci-après au 8.5.7, qui font partie du canal. Ils peuvent avoir à effectuer des conversions de format ou de protocole ou à fournir des vérifications d'ordre comptable ou de contrôle d'accès. La construction d'un intercepteur peut s'envisager à partir d'objets de protocole, d'objets lieurs et de talons, selon la nature des fonctions qu'il doit remplir.

Pour des raisons de simplicité, le dessin de la Figure 4 représente une configuration dans laquelle un intercepteur, des talons et des objets lieurs et de protocole ne forment qu'un seul canal entre deux objets d'ingénierie de base. Mais, d'une manière plus générale, ce genre de configuration est capable de prendre en charge des canaux reliant une pluralité de couples d'objets d'ingénierie (voir la Figure 5), ou de canaux à points d'extrémités multiples adaptés au cas de diverses formes de communications de groupes (voir la Figure 6) ou de communications multiples. Dans le cas de la Figure 5, les objets lieurs portent la responsabilité de la coordination de la communication, alors que les mécanismes de communication multiple peuvent être fournis, en fonction des moyens techniques disponibles, soit par les objets lieurs, soit par les objets de protocole. Les canaux à extrémités multiples servent à la transparence à la duplication.

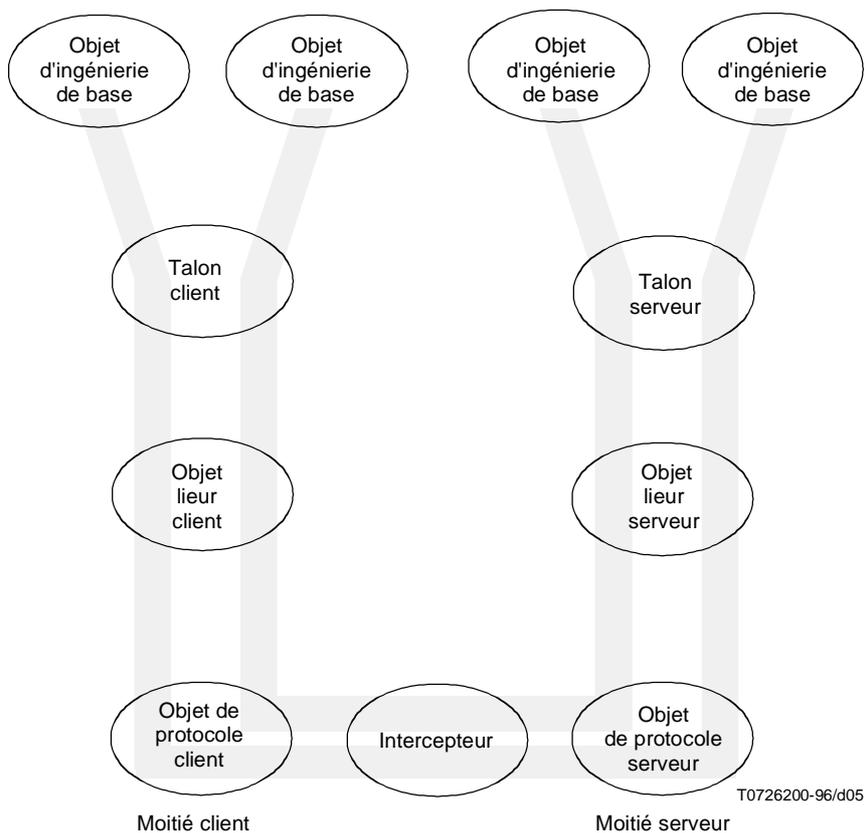


Figure 5 – Exemple de configuration à canaux multiples

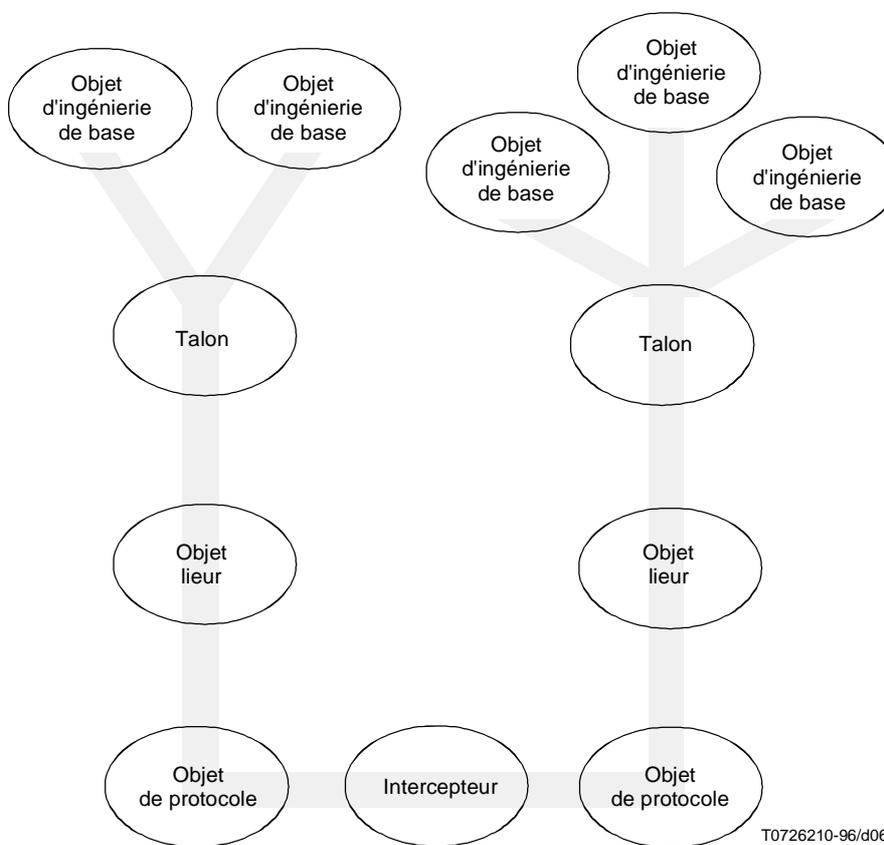


Figure 6 – Exemple de configuration à canal de groupe

### 8.5.3 Références d'interface

Lors de sa création, une interface reçoit une référence d'interface produite à cet effet. Le noyau participe à ce processus afin que la référence soit dénuée d'ambiguïté. Des ressources sont allouées et initialisées en quantité suffisante pour que les objets d'ingénierie du nœud puissent participer à des liaisons au cas où ils seraient appelés à le faire.

La référence d'interface est la clé d'accès à une grande quantité d'informations. Par cette référence, on peut découvrir le type de l'interface, une adresse de communication à laquelle il est possible de lancer des liaisons avec elle, ainsi que d'autres informations touchant aux comportements qu'il est légitime d'attendre des talons, des objets lieurs et des objets de protocole du canal, toutes informations nécessaires au succès des liaisons futures. C'est aussi le point de départ pour l'appel aux fonctions de traitement des erreurs; la connaissance d'une référence d'interface permet de contacter un *relocalisateur* approprié.

Mais ceci n'implique pas que toute l'information soit codée dans la référence d'interface, ce qui en ferait un très gros article à manipuler. L'exigence architecturale est qu'il existe un moyen d'atteindre l'information requise à partir de la référence d'interface, mais la méthode exacte utilisée, en termes d'interrogation et de décodage d'autres objets d'ingénierie, reste une variable de conception du système.

Il peut exister, en plus de ces variations de conception, d'autres variations qui résultent de la présence d'une pluralité de domaines de désignation par rapport auxquels est effectuée l'allocation des références. C'est pour ces raisons que les intercepteurs, ou d'autres objets d'ingénierie du canal, peuvent être amenés à transformer les références d'interfaces lorsqu'elles traversent des frontières de domaines.

### 8.5.4 Liaison

Il existe deux sortes de liaisons d'ingénierie. A l'intérieur d'une grappe ou entre les objets d'ingénierie qui coopèrent au sein d'un nœud à la fourniture d'un canal, il s'agit de liaisons locales que produisent des mécanismes particuliers au système. L'architecture les considère comme des liaisons primitives. Quant aux liaisons appuyées sur des canaux, qui fournissent des transparences à la répartition, elles sont appelées liaisons réparties. En général, leur création demandera des interactions entre un certain nombre de nœuds pour l'établissement du canal.

### 8.5.5 Etablissement d'un canal

Pour pouvoir établir une liaison entre des objets d'ingénierie, il est nécessaire d'identifier et de décrire les interfaces de ces objets. La référence d'interface donne à cet égard des informations assez détaillées pour pouvoir lier l'interface. Cette référence est transmissible en une ou plusieurs interactions entre objets d'ingénierie, ce qui permet à tout objet qui reçoit une référence d'interface de déclencher l'établissement d'une liaison avec cette interface sans avoir besoin de plus d'information.

Un objet d'ingénierie qui désire passer une référence d'interface à l'une de ses interfaces demande à son noyau, par le truchement d'une opération sur son interface de gestion, de créer une référence d'interface. La référence d'interface donne l'identité de l'interface et toute l'information nécessaire pour créer une liaison avec elle.

Le type d'une interface d'ingénierie fait connaître le type d'interface de traitement qu'elle met en œuvre ainsi que la configuration, en termes de talons, d'objets lieurs, d'objets de protocole et d'intercepteurs (décrits ci-après au 8.5.7), du canal sur laquelle elle s'appuie. Lorsque doivent être liées ensemble des interfaces d'ingénierie qui assument des types compatibles d'interfaces de traitement, la configuration exacte du canal capable de prendre en charge toutes les interfaces qui y sont incorporées est négociée par la procédure d'établissement de la liaison. Il arrive que cette configuration puisse être optimisée. Si par exemple les interfaces sont capables de traiter la même représentation des données, il n'est pas nécessaire que le canal comprenne un objet d'ingénierie de conversion entre représentations de données, bien qu'il reste nécessaire d'organiser une copie des arguments ou des résultats sous forme de message.

Les liaisons entre interfaces sont créées par interaction entre des objets noyaux (en général aux nœuds où sont localisées les interfaces). Quand donc un noyau crée une référence d'interface, il donne des instructions sur la manière de le contacter pour qu'il établisse une liaison avec l'interface référencée. En termes de langage d'ingénierie, le noyau désigne une ou plusieurs *interfaces de communication* où doit se produire l'interaction entre noyaux. L'identité d'une interface de communication est susceptible de comporter des informations relatives aux protocoles de communication que cette interface met en œuvre.

Normalement, le noyau désigne l'une de ses propres interfaces de communication pour servir aux interactions d'établissement de liaisons, mais une approche plus générale est aussi permise. L'interface de communication, associée à son protocole de communication, qui sert lors de l'interaction d'établissement de liaison, n'emporte aucune implication quant à l'interface de communication, ni au protocole de communication, qui serviront à la liaison après sa création. L'utilisation d'une conversation téléphonique pour préparer l'envoi de documents papier par le système postal peut constituer une analogie à cette situation.

La référence d'interface transmet une information éminemment compliquée que tout l'ensemble du système ODP est en mesure d'interpréter. La structure détaillée d'une référence d'interface fait par conséquent l'objet d'une normalisation particulière. Le modèle de référence ne spécifie pas si la référence d'interface contient physiquement toute l'information décrite ici, ou si elle n'est qu'une clé d'accès à cette information par interaction avec d'autres objets d'ingénierie (pour, par exemple, la résolution d'un nom par un serveur de noms).

Comme exemple d'établissement de canal, prenons l'établissement d'un canal flux entre deux objets d'ingénierie. Plusieurs étapes interviennent.

**\* Etape 1**

L'un des deux objets d'ingénierie déclenche la formation d'un canal par interaction avec son noyau. La syntaxe d'interaction pourrait se présenter sous la forme suivante:

**InitChannel** (*StreamChannel*, *producer/consumer*, *IFPC1*, **result** *IFrefStreamchannel*)

où *StreamChannel* (canal flux) est du type "*StreamChannel*" et *Streamchannel* est le type de canal à créer; *producer/consumer* (producteur et consommateur) indique que l'objet d'ingénierie en question jouera sur ce canal les deux rôles de producteur et de consommateur; *IFPC1* est l'interface de l'objet d'ingénierie à lier au canal flux.

Lorsque se produit cette interaction, le noyau crée un objet talon, un objet lieu et un objet de communication qui correspondent au type du canal et à son rôle. Ces objets d'ingénierie sont liés pour créer la première partie d'un canal flux. L'interface de présentation de l'objet talon est liée à l'interface *IFPC1*. Puis l'objet talon est lié à l'objet lieu qui est lié à l'objet de protocole. Il résulte de cette interaction la référence d'interface (*IFrefStreamchannel*). Celle-ci sera communiquée aux objets d'ingénierie qui veulent se lier au canal.

**\* Etape 2**

La référence d'interface du canal est communiquée au second objet d'ingénierie. Celui-ci entre en interaction avec son noyau pour se lier au canal par le biais de l'interaction suivante:

**BindChannel** (*StreamChannel*, *producer/consumer*, *IFPC2*, *IFrefStreamchannel*)

où *StreamChannel* est le type de canal; *producer/consumer* indique que le second objet d'ingénierie jouera sur le canal flux les deux rôles de producteur et de consommateur; *IFPC2* est l'interface du second objet à lier au canal flux.

Le noyau détermine à partir de la référence d'interface *IFrefStreamchannel* le type de canal et la localisation des objets de protocole des autres participants au canal flux. Il crée un objet talon, un objet lieu et un objet de protocole qui correspondent au type de canal des autres participants et au rôle. Ces objets sont liés à la partie déjà établie du canal et au second objet d'ingénierie. Les objets lieux du canal entrent alors en interaction l'un avec l'autre afin de permettre de communiquer au travers du canal.

**\* Etape 3**

D'autres objets peuvent se lier au canal existant par emploi de la même interaction **BindChannel** ().

### 8.5.6 Interfaces de gestion

Seul un objet peut modifier son propre comportement. Un objet peut répondre à des demandes de modification de son comportement émises par une application de gestion et peut, en conséquence, déléguer la responsabilité d'une partie de sa gestion à l'application de gestion.

La gestion des ressources exige la faculté d'invoquer des opérations de gestion sur les services pris individuellement, sur les objets d'ingénierie qui contiennent les services, sur la grappe qui contient les objets d'ingénierie, sur la capsule qui contient la grappe et sur le nœud qui héberge la capsule.

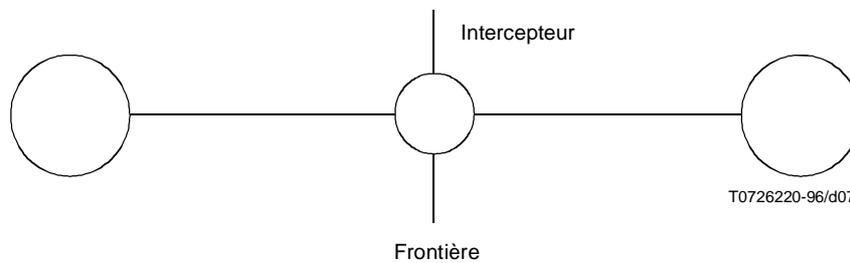
### 8.5.7 Intercepteurs

La spécification d'intercepteurs répond à un besoin de fédération de domaines techniques ou administratifs précisé par une spécification d'entreprise. La notion d'intercepteur correspond à celle d'objets "portails" (*gateway*), "agents" ou "de surveillance" qui se situent entre deux domaines dont ils permettent ou autorisent les interactions sur la base d'un contrat passé entre les Administrations pour préciser les conditions de leur fédération.

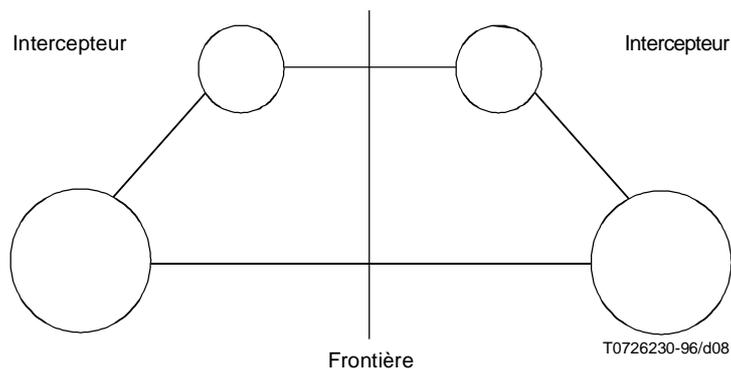
Au sein d'un domaine technique, les objets noyaux voient de manière identique les représentations des données, les fonctions des protocoles, la désignation et l'adressage. La rencontre de deux domaines techniques ouvre une opportunité de fusion des domaines, qui demande que les objets de traitement de chaque domaine soient élargis pour pouvoir employer les moyens techniques du domaine étranger en plus de ceux de leur propre domaine, à moins que tous deux ne passent à des solutions techniques communes plus générales. Lorsque ce cas se produit, la frontière technique s'évanouit et il suffit de transparence d'accès. Les intercepteurs interviennent dans le cas où il n'est pas possible de modifier les solutions techniques utilisées dans l'un ou l'autre des systèmes. Il faut alors agir à la frontière entre les systèmes pour convertir les protocoles et traduire les noms.

Ces dernières activités sont exercées aux frontières techniques par des intercepteurs en ligne qui sont impliqués dans toutes les interactions qui traversent la frontière. Pour des raisons d'efficacité, il ne serait utilisé qu'un seul intercepteur par frontière technique (voir la Figure 7).

Les intercepteurs de frontières administratives appartiennent totalement à une Administration pour laquelle ils exercent des responsabilités de protection. Un tel intercepteur pourrait servir, par exemple, à traduire les informations de sécurité qui contiennent les autorisations. Il pourrait être utilisé avant une interaction par l'infrastructure de l'objet de traitement demandeur. Il est permis à un intercepteur administratif de communiquer avec des intercepteurs similaires appartenant à d'autres domaines pour échanger des informations comme des clés de chiffrement et pour vérifier les informations administratives avant de les traduire (voir la Figure 8). Les traductions effectuées par les intercepteurs administratifs restent utilisables au cours de plusieurs interactions successives entre objets de traitement, sans nouvel appel à l'intercepteur.



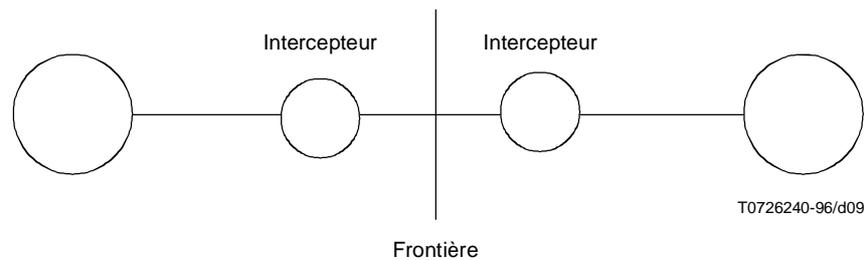
**Figure 7 – Intercepteur en ligne – Frontière technique**



**Figure 8 – Intercepteur partagé – Frontière administrative**

En cas de coexistence d'une frontière technique avec une frontière administrative, il faudrait faire appel aux deux types d'intercepteurs (en combinant les Figures 7 et 8). On peut aussi partager l'intercepteur en ligne pour obtenir deux intercepteurs en ligne, un par domaine administratif, afin de prendre en charge la relation de confiance (voir la Figure 9).

Les frontières comportent en général N voies, car il peut arriver que plus de deux sous-systèmes se rencontrent au même endroit. Mais comme une frontière à N voies peut toujours être traitée logiquement comme N frontières à deux voies, seul doit être traité le cas d'intercepteurs à deux voies.



**Figure 9 – Intercepteur partagé – Frontière technique et administrative combinée**

Il advient des conditions spéciales lorsque la fédération implique une liaison entre les courtiers des deux domaines. Si un courtier est accessible au travers d'un intercepteur en ligne, ce courtier peut alors servir à accéder à d'autres courtiers. L'intercepteur doit cependant fournir des fonctions spéciales d'initialisation qui donnent accès au travers de lui-même à au moins un courtier de départ dans les deux directions. Il est possible de réaliser cette initialisation en faisant fédérer par l'intercepteur les courtiers situés des deux côtés de la frontière, en faisant de l'intercepteur un courtier mandataire ou en établissant un courtier au sein de l'intercepteur lui-même.

### 8.5.8 Points de conformité

La structuration de la spécification d'ingénierie en grappes, capsules et nœuds et l'usage de canaux structurés pour la prise en charge des interactions donnent naissance à une grande quantité d'interfaces, toutes susceptibles d'être choisies comme points de conformité où exercer des activités d'observation et de vérification de la conformité.

Les diverses interfaces mettent en jeu des conformités de différentes sortes. L'interface entre objets de protocole est un point de conformité d'interfonctionnement où peuvent intervenir des méthodes familières fondées, comme les techniques de vérification de l'OSI, sur l'observation du comportement de la communication. La plupart des autres interfaces sont internes à un nœud et représentent des frontières entre modules logiciels. Il s'agit de *points de référence de programmation* où peuvent être vérifiées la compatibilité et la portabilité des logiciels. Il se peut que certaines des interfaces des objets d'ingénierie de base autorisent d'autres formes d'essais, pour la conformité d'échange ou physique (vérification de la correction des interactions avec le monde réel). Elles peuvent aussi servir de points de conformité pour l'évaluation de la cohérence du comportement avec les exigences exprimées dans les spécifications d'entreprise et d'information.

## 8.6 Langage de technologie

La spécification de technologie décrit la réalisation du système ODP en termes de configuration d'objets techniques qui en représentent les composants matériels et logiciels. Elle est soumise à des contraintes de coût et de disponibilité des objets techniques matériels et logiciels susceptibles de répondre à la spécification. Ils peuvent se conformer à des normes de réalisation qui constituent en fait des gabarits pour les objets techniques. Le point de vue technologie fournit ainsi un lien entre l'ensemble des spécifications de points de vue et la réalisation effective en donnant la liste des normes utilisées pour fournir les opérations de base que requièrent les autres spécifications de points de vue. Le but de la spécification de technologie est d'apporter, par le choix de solutions normalisées pour la réalisation des composants de base et des mécanismes de communication, l'information additionnelle que demandent la mise en œuvre et les essais. Ce choix, qui est nécessaire pour finaliser la spécification du système, est largement dissocié du reste du processus de conception.

Le choix des solutions techniques a pourtant des conséquences. Un des domaines où ce choix se répercute sur d'autres aspects de la conception du système est celui de la recherche d'un niveau donné de qualité de service. Les choix effectués dans le cadre du point de vue technologie déterminent le coût d'exécution des interactions et, indirectement par conséquent, le niveau de qualité de service que l'on peut attendre d'un comportement défini par les spécifications d'autres points de vue.

La spécification de technologie joue un rôle très important dans le processus de vérification de la conformité. Elle détermine les points de conformité du système réel où le testeur a la possibilité de procéder à des observations; elle donne l'information nécessaire pour réussir à interpréter les observations que peut faire un testeur dans les termes des vocabulaires et des concepts dont se servent les autres points de vue de spécification du système. Elle permet par exemple de reconnaître les interactions valides en sorte qu'il soit possible d'en vérifier l'à-propos par rapport au comportement de certains objets de technologie bien déterminés. Les informations demandées à cet effet sont appelées IXIT, informations complémentaires nécessaires aux essais de conformité d'une implémentation (*implementation extra information for testing*).

## 8.7 Cohérence des points de vue

Les cinq spécifications de points de vue relatives à un système sont reliées par des déclarations qui définissent les relations existant entre les termes clés qu'elles comportent. Elles entraînent que:

- les spécifications se rapportent à un système unique et ne sont pas indépendantes les unes des autres;
- les spécifications présentent une cohérence interne;
- le comportement observable aux points de conformité de la spécification de technologie peut être relié à des exigences exprimées dans les autres spécifications de point de vue.

En conséquence des correspondances qui existent entre les noms, un grand nombre de ces liens dériveront implicitement des notations employées. Mais il faudra cependant expliciter quelques-unes des contraintes essentielles. Dans l'architecture, des contraintes portant sur les relations entre les termes utilisés dans les langages eux-mêmes restreignent les mappages permis. La plupart des contraintes s'appliquent entre des termes des langages de traitement et d'ingénierie. Leur définition est conçue pour conduire à des interprétations cohérentes lorsque sont spécifiés séparément des composants du système comme ceux qui prennent les fonctions ODP en charge.

Il est nécessaire de disposer de mappages clairs entre les points de vue pour que des outils automatiques de développement puissent fournir les processus d'identification des interfaces et de fourniture des transparences. Par exemple: on peut réaliser un objet de traitement par un jeu d'objets d'ingénierie mutuellement liés, mais un objet d'ingénierie unique ne peut pas représenter plusieurs objets de traitement; une interface de traitement ne peut pas se subdiviser entre des interfaces d'ingénierie distinctes, sauf si ces dernières sont liées par des fonctions de duplication; des identifiants d'ingénierie identifient sans ambiguïté les interfaces de traitement. Des contraintes de ce genre servent à garantir que des mécanismes communs d'ingénierie seront en mesure de prendre en charge toute l'étendue des comportements de traitement possibles.

Afin d'illustrer la nature des correspondances susceptibles de s'appliquer, imaginons par exemple qu'une spécification particulière d'un système donné puisse être représentée sous le point de vue 1 (PV1) sous forme d'un ensemble d'objets de PV1 en interaction, comme le montre la Figure 10.

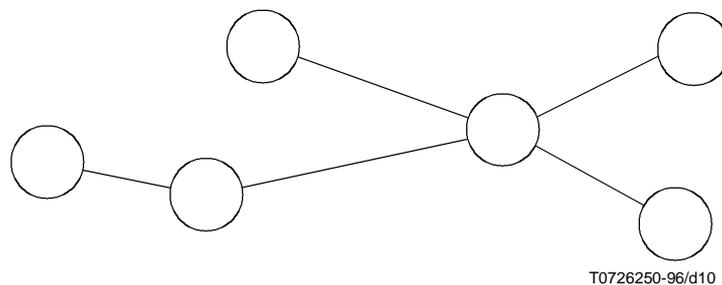


Figure 10 – Vue d'un système sous PV1

Sous un autre point de vue, (PV2), le même système pourrait être représenté sous la forme d'un ensemble différent d'objets différents de PV2 en interaction, comme le montre la Figure 11.

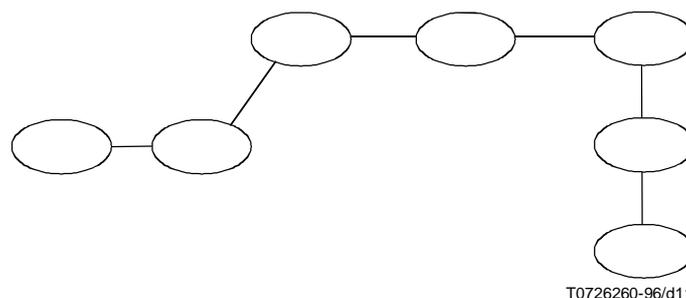


Figure 11 – Vue du système sous PV2

Comme il s'agit de deux descriptions du même système, il est possible de regrouper les objets des deux figures précédentes afin de vérifier que les règles de correspondance entre les groupes sont satisfaites. La Figure 12 illustre ce processus.

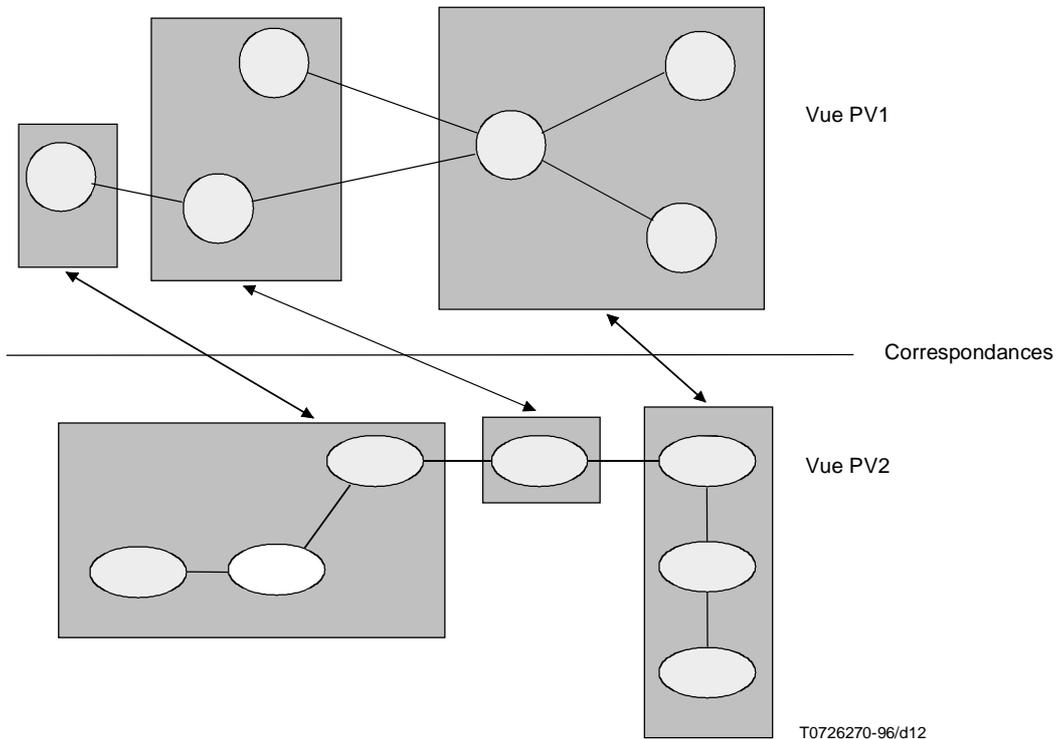


Figure 12 – Correspondances entre différents points de vue d'un système

S'il n'est pas possible d'établir de correspondance de cette sorte, c'est que les deux descriptions ne sont pas cohérentes. Il y a lieu de les affiner jusqu'à ce qu'il soit possible de montrer une correspondance.

Les configurations d'objets à comparer (les configurations des boîtes de la Figure 12) sont en général définies à la seule fin de trouver une correspondance entre deux spécifications.

En d'autres termes, pour comparer une spécification SpecA écrite dans un langage de point de vue L1 avec une autre spécification SpecC du même système écrite dans un autre langage de point de vue, L2, il faut dans le cas général:

- a) transformer SpecA en une autre spécification en L2. Soit SpecB cette spécification. Notez que le modèle de référence ODP **ne définit pas** d'algorithmes de transformation;

NOTE – Pour mener à bien cette transformation, il est quelquefois pratique, pour mieux vérifier les correspondances, de dériver de SpecA une autre spécification du même système, toujours dans le langage L1, équivalent à SpecA. C'est par exemple le regroupement dans d'autres objets (les boîtes) des objets définis dans le système de la Figure 12.

- b) vérifier qu'il n'existe pas de conflits entre SpecB et SpecC.

Les correspondances s'appliquent entre des spécifications exprimées par des langages de points de vue différents et non entre les termes de ces langages. Autrement dit, il n'existe pas de traduction directe d'un langage de point de vue dans un autre.

### 8.7.1 Cohérence du point de vue entreprise avec les autres points de vue

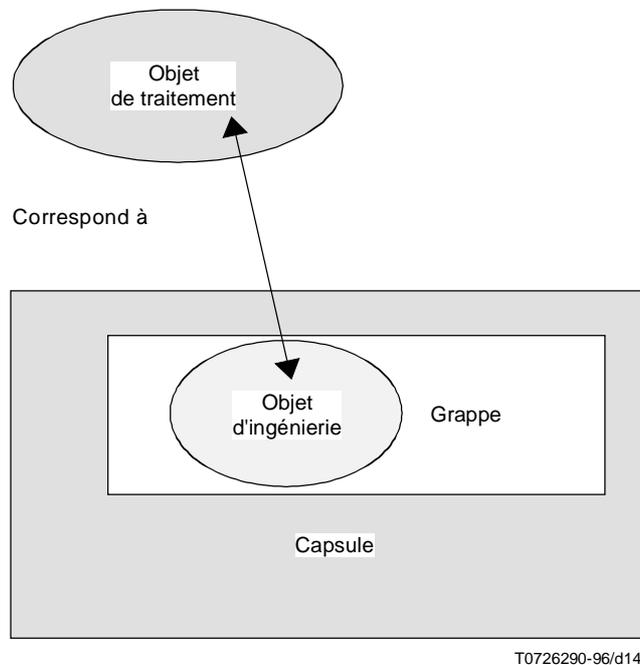
Le langage d'entreprise devrait servir de base pour la spécification des buts de l'entreprise que toutes les autres spécifications de point de vue doivent refléter, explicitement ou implicitement. Le point de vue entreprise décrit explicitement les objectifs du système en termes de sociétariat, de rôles, d'actions, d'intentions, d'usage et de politiques.



L'objet lieu de la spécification de traitement représentée à la Figure 13 correspond à une configuration de canal dans la spécification d'ingénierie. Les contraintes d'environnement (sécurité, qualité de service, etc.) qui s'appliquent aux interfaces ainsi liées sont prises en compte lors de l'établissement du canal entre les objets d'ingénierie de base concernés.

L'interface de commande de l'objet lieu dans la spécification de traitement correspond aux interfaces des talons et des objets lieux des différents nœuds de la spécification d'ingénierie. On pourrait introduire un objet de contrôle du canal, chargé de lancer les opérations de commande. La communication entre l'objet de contrôle du canal, d'une part, et les talons et objets lieux, d'autre part, fonctionne au travers de canaux établis à cet effet et que ne montre pas la Figure 13. Il peut être créé un ensemble d'objets d'ingénierie en support, comme des objets de synchronisation, pour gérer et régir un ensemble de canaux qui entretiennent des relations mutuelles.

D'une manière générale, une spécification de traitement décrit les fonctions d'un système sous la forme d'un ensemble d'objets de traitement en interaction mutuelle. La spécification d'ingénierie se trouve contrainte par la spécification de traitement en ce qu'elle doit en respecter les objets de traitement et leurs interfaces. Chaque objet de traitement doit être représenté par un objet d'ingénierie de base ou par un groupe d'objets d'ingénierie de base. Dans le cas le plus simple, à un objet de traitement correspond un objet d'ingénierie unique. Par exemple, après compilation un programme d'application correspond à un objet d'ingénierie à charger comme module de chargement dans une grappe après édition des liens avec d'autres objets d'ingénierie, talons, objets lieux, etc. La Figure 14 illustre cette configuration.



**Figure 14 – Projection de un sur un**

A chaque interface de traitement doit correspondre une interface d'ingénierie, sauf en cas de transparences impliquant des copies d'objets. Dans ce cas, la même interface de traitement peut être associée à différents identifiants d'interface d'ingénierie, ce qui permet d'exercer la fonction de duplication pour des raisons, par exemple, de performances. L'interface de traitement est associée à un jeu de *références d'interfaces d'ingénierie* qui correspondent aux différents objets d'ingénierie. Il est nécessaire, pour que le système maintienne un état global cohérent, de coordonner les activités de ces objets d'ingénierie en faisant appel à des objets de duplication. La Figure 15 donne un exemple de ce cas. Elle montre deux objets d'ingénierie de base situés dans deux nœuds différents qui dupliquent les fonctions offertes à une interface de traitement. Comme les objets d'ingénierie sont dans des nœuds différents, la coordination de la duplication est assurée par deux objets de duplication, un par nœud, qui communiquent au travers d'un canal.

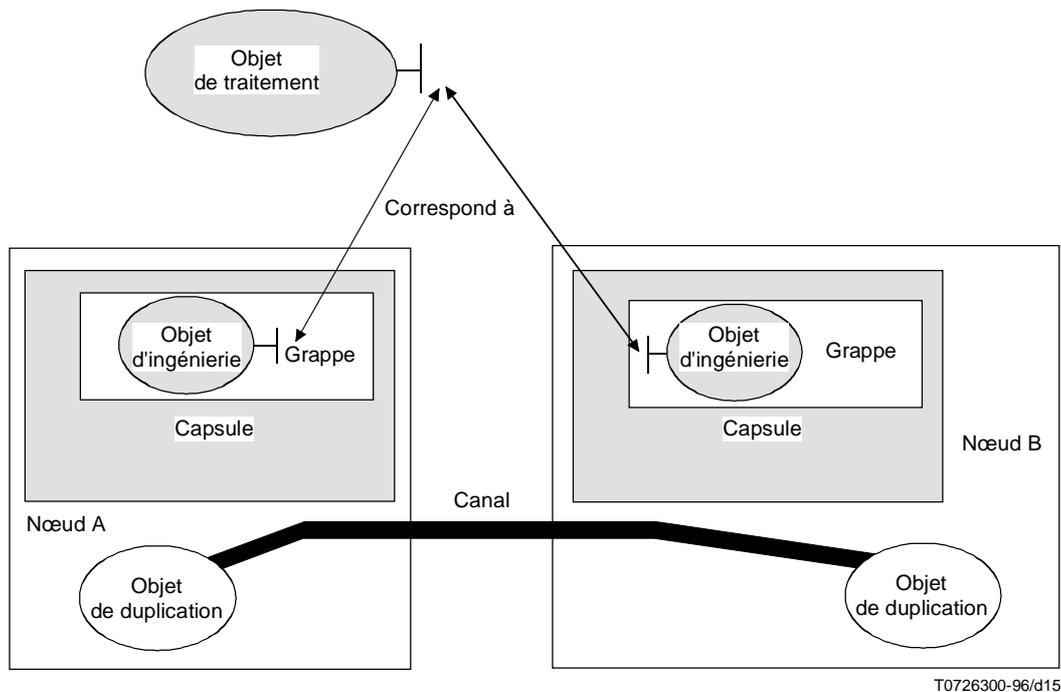


Figure 15 – Projection de un sur plusieurs

Le processus d'affinement entre la spécification de traitement et la spécification d'ingénierie peut se limiter à la simple détermination des objets de soutien qui conviennent à la construction des canaux qui représentent les objets lieux de la spécification de traitement. Il peut être nécessaire, dans d'autres cas, de procéder à d'importantes transformations des gabarits des objets de traitement eux-mêmes pour remplacer des déclarations relatives au comportement (contraintes de synchronisation par exemple) par l'emploi explicite de fonctions ODP adéquates (la fonction de transaction par exemple).

Si l'on dispose d'une connaissance suffisante de la configuration du système, il est possible, dans les cas où le caractère de généralité de la structure d'ingénierie n'est pas complètement requis, de remonter des optimisations particulières. C'est ainsi que le réalisateur peut court-circuiter des interactions entre objets d'ingénierie de base du même nœud, en utilisant par exemple des procédures plus performantes d'appels de procédures locales, du moment que ces courts-circuits n'affectent pas l'interfonctionnement au travers d'interfaces visibles avec des objets situés dans d'autres noyaux. L'architecture est ainsi capable de venir à bout de la distribution sans exiger un grand nombre de mécanismes pour pourvoir aux interactions locales et distantes et sans pour autant sacrifier l'efficacité d'exécution.

## 8.8 Fonctions ODP

La Rec. UIT-T X.903 | ISO/CEI 10746-3 donne les descriptions succinctes d'un jeu de fonctions ODP. Ces fonctions ou bien jouent un rôle fondamental ou bien sont d'application courante dans la construction des systèmes ODP. Les spécifications détaillées de ces fonctions feront l'objet d'activités de normalisation spécifiques. Il sera permis aux normes qui en résulteront de combiner des spécifications de fonctions ODP pour spécifier les composants des systèmes ODP.

L'ensemble complet des fonctions ODP se répartit en quatre groupes:

- a) fonctions de gestion;
- b) fonctions de coordination;
- c) fonctions de conteneur;
- d) fonctions de sécurité.

### 8.8.1 Fonctions de gestion

Les fonctions de gestion comprennent:

- la fonction de gestion de nœud;
- la fonction de gestion d'objet;

- la fonction de gestion de grappe;
- la fonction de gestion de capsule.

La fonction de gestion de nœud est fournie par le noyau d'un nœud. Elle s'intéresse aux fonctions de traitement, de mise en mémoire et de communication du nœud. Elle pourvoit:

- à la gestion des fils d'exécution;
- à la gestion des accès à l'horloge et du temporisateur;
- à la création des canaux et à la manipulation des références d'interfaces d'ingénierie;
- à l'instanciation des gabarits de capsules et à la suppression des capsules.

Tout objet peut assumer, lorsqu'elle est demandée, la fonction de gestion d'objet. Elle permet la pose de points de reprise et la suppression de l'objet.

La fonction de gestion de grappe est fournie par un gestionnaire de grappe. Elle permet la pose de points de reprise, la reprise, la migration, la désactivation ou la suppression de la grappe.

La fonction de gestion de capsule est fournie par un gestionnaire de capsule. Elle permet l'instanciation (y compris la reprise et la réactivation), la pose de points de reprise, la désactivation ou la suppression de toutes les grappes d'une capsule et de la capsule elle-même.

### 8.8.2 Fonctions de coordination

Les fonctions de coordination comprennent:

- la fonction de notification d'événement;
- la fonction de point de reprise et de reprise;
- la fonction de désactivation et de réactivation;
- la fonction de groupe;
- la fonction de duplication;
- la fonction de migration;
- la fonction de transaction;
- la fonction de ramasse-miettes pour les références d'interfaces d'ingénierie.

La fonction de notification d'événement enregistre les historiques des événements, qu'elle rend disponibles. Les producteurs d'événements sont en interaction avec cette fonction pour la création de l'historique d'événement; la fonction informe des consommateurs d'événements, préalablement enregistrés, de la disponibilité des historiques d'événements.

La fonction de point de reprise et de reprise coordonne la pose de points de reprise dans les grappes et la reprise de grappes défaillantes à partir de *points de reprise*. Elle est régie par des politiques qui spécifient quand il y a lieu de poser des points de reprise dans les grappes et où doivent être conservés les points de reprise correspondants, quand et où doivent se produire les reprises des grappes et sur quels points il y a lieu d'exécuter des reprises.

La fonction de désactivation et de réactivation coordonne les désactivations et réactivations de grappes. Elle est régie par des politiques qui spécifient quand il y a lieu de désactiver les grappes et où doivent être conservés les points de reprise correspondants, quand et où doivent se produire les réactivations et quels points de reprise doivent servir à la réactivation.

La fonction de groupe fournit les mécanismes nécessaires à la coordination des objets qui interviennent dans une liaison à plusieurs partenaires.

La fonction de duplication traite du cas particulier d'un groupe au sein duquel les objets exhibent une compatibilité de comportement. Elle fournit les mécanismes grâce auxquels le groupe apparaît aux autres objets comme un objet unique. Elle permet aussi d'accroître et de réduire le sociétariat du groupe. Cette fonction peut s'utiliser au niveau d'une grappe, en conjonction avec la fonction de groupe, pour constituer un ensemble coordonné de groupes de duplication où les objets de chaque grappe forment un groupe de duplication.

La fonction de migration coordonne la migration d'une grappe d'une capsule à une autre. Elle peut fonctionner soit par duplication de la grappe, faisant alors usage de la fonction de duplication, soit par désactivation de la grappe suivie de réactivation dans une autre capsule, faisant alors appel à la fonction de désactivation et de réactivation.

La fonction de transaction coordonne et gère un groupe de transactions pour atteindre un niveau spécifié de visibilité et de permanence, sous la dépendance de politiques qui décident quelles sont les actions qui intéressent la transaction. Une transaction ACID est un cas particulier de la fonction de transaction dans lequel les transactions sont douées des propriétés d'atomicité, de cohérence, d'isolation et de durabilité.

La fonction de ramasse-miettes pour les références d'interfaces d'ingénierie surveille le transfert de références d'interfaces d'ingénierie entre objets d'ingénierie appartenant à des grappes différentes pour déterminer le moment à partir duquel il n'y a plus besoin de l'infrastructure qui prend la référence en charge parce qu'il n'existe plus aucun objet dans aucune grappe qui soit en mesure d'établir de liaison avec l'interface en question.

### 8.8.3 Fonctions de conteneur

Les fonctions de conteneur comprennent:

- la fonction de stockage;
- une fonction de gestion de base d'information;
- la fonction de relocalisation;
- la fonction de conteneur de types;
- la fonction de courtage.

La fonction de stockage met les données en mémoire.

La fonction de gestion de base d'information gère un conteneur de l'information décrite par un schéma d'information. Elle permet de modifier et de mettre à jour aussi bien le schéma que le conteneur, ainsi que d'interroger le conteneur.

La fonction de relocalisation gère un conteneur de positions d'interfaces et de fonctions de gestion pour les grappes qui prennent ces interfaces en charge.

La fonction de conteneur de types gère un conteneur de spécifications de types et de relations entre types.

La fonction de courtage donne le moyen aux prestataires de services d'exporter des *offres de service* sous la forme d'informations sur l'interface où est fourni un service et aux utilisateurs de services d'importer les offres de service qui concordent avec leurs exigences particulières.

### 8.8.4 Fonctions de sécurité

Les fonctions de sécurité traitent des exigences relatives à la confidentialité, à l'intégrité, à la disponibilité et à la capacité de tenir des comptes. Elles comprennent:

- la fonction de contrôle d'accès;
- la fonction d'audit de sécurité;
- la fonction d'authentification;
- la fonction d'intégrité;
- la fonction de confidentialité;
- la fonction de non-répudiation;
- la fonction de gestion de clés.

La fonction de contrôle d'accès empêche les interactions non autorisées avec un objet.

La fonction d'audit de sécurité surveille les actions relatives à la sécurité, recueille des informations à leur sujet et permet d'analyser ces informations pour en tirer une évaluation des politiques, des contrôles et des procédures de sécurité.

La fonction d'authentification garantit l'identité que prétend posséder un objet.

La fonction d'intégrité détecte ou prévient toute création, altération ou suppression non autorisée des données.

La fonction de confidentialité prévient toute révélation non autorisée d'information.

La fonction de non-répudiation prévient le déni par un objet de son implication dans l'interaction.

La fonction de gestion de clés donne les outils nécessaires à la gestion des clés cryptographiques.

Ces fonctions fournissent des services qui sont applicables tant aux objets eux-mêmes qu'aux interactions entre objets. Les mécanismes de sécurité eux-mêmes ont besoin de protection puisque des menaces intelligentes et malignes sont une des caractéristiques des environnements qu'il est nécessaire de sécuriser. L'encapsulation d'ingénierie peut participer à la fourniture de cette protection. Dans de nombreux cas, il est possible d'offrir des services de sécurité sans avoir besoin de recourir à une référence dans les spécifications de traitement.

## 8.9 Transparences ODP à la répartition

Comme expliqué au 8.5, les objets d'ingénierie interagissent les uns avec les autres par l'intermédiaire de talons, d'objets lieurs, d'objets de protocole, d'intercepteurs et de noyaux. Les objets d'ingénierie coopèrent dans le but de fournir une transparence en rendant uniforme un certain aspect de la répartition des objets d'ingénierie de base qu'ils prennent en charge. Ils peuvent par exemple traduire des invocations en échanges de messages faisant appel à des formats communs de données pour masquer les différences dans le codage des données.

Certaines formes de transparence font appel à des fonctions de soutien. Si par exemple les objets d'ingénierie peuvent se déplacer d'un endroit à l'autre, il faut disposer d'un moyen d'enregistrement et de découverte de la position courante d'un composant. C'est la fonction de relocalisation.

Les fonctions de soutien sont elles-mêmes susceptibles d'éprouver des besoins de transparence. La disponibilité d'une fonction de relocalisation, par exemple, peut être accrue par usage de la duplication.

Les fonctions de soutien sont représentées dans le modèle sous forme d'objets d'ingénierie, grâce à quoi l'architecture peut faire preuve d'un très haut niveau de souplesse de configuration et de réemploi des concepts architecturaux pour définir la répartition de ces fonctions. Dans une réalisation, elles peuvent par exemple se trouver soit rassemblées pour améliorer les performances, soit dupliquées pour parfaire la fiabilité.

La transparence à la répartition est la propriété grâce à laquelle les conséquences de la répartition sont dissimulées aux utilisateurs finals et aux concepteurs de systèmes aux niveaux des langages d'entreprise, d'information et de traitement. Des normes de "composition de composants" contiendront des recettes précises pour élaborer l'offre des transparences par l'emploi des fonctions et des composants de base.

### 8.9.1 Transparence d'accès

La transparence d'accès rend possible l'interfonctionnement des ordinateurs en dépit d'architectures et de langages de programmation hétérogènes.

Cette transparence joue un rôle essentiel dans la construction de systèmes répartis employant des éléments hétérogènes, architectures des ordinateurs, langages de programmation, etc.

### 8.9.2 Transparence aux défaillances

La transparence aux défaillances *cache* à un objet de traitement la défaillance et, éventuellement, la reprise d'autres objets de traitement, ou de lui-même, pour qu'il fasse preuve de résistance aux pannes. Elle peut être fournie par une infrastructure adéquate ou, sinon, par la fonction de point de reprise et de reprise ou par la fonction de duplication associée à la fonction de relocalisation.

Un service qui, pour fournir la transparence aux défaillances, s'appuierait sur la seule fonction de point de reprise, doit fournir à ses clients, au titre du service lui-même, les moyens qui leur permettront de détecter qu'il s'est produit un redémarrage à partir d'un point de reprise et que l'information d'état que détient le client sur le service risque d'être périmée. S'il existe un besoin de cohérence entre plusieurs objets de traitement, alors devrait être spécifiée la transparence aux transactions.

### 8.9.3 Transparence à la position

La transparence à la position cache à un objet de traitement la position dans l'espace de chacun des objets avec lesquels il est en interaction. Cette propriété implique qu'il soit possible d'identifier les interfaces et d'y accéder sans avoir à en spécifier la localisation spatiale.

### 8.9.4 Transparence à la migration

La transparence à la migration cache à un objet de traitement le fait qu'il a été déplacé. Elle repose sur la fonction de migration.

Il est possible de combiner la transparence à la migration avec la transparence à la persistance ou avec la transparence aux défaillances en sorte qu'une grappe ne soit pas réactivée sur son site d'origine mais que le gabarit de grappe soit transféré directement au nouvel emplacement, où la grappe sera réactivée.

### 8.9.5      **Transparence à la persistance**

La transparence à la persistance cache aux objets de traitement l'allocation et le retrait des ressources nécessaires aux grappes ou à leurs gabarits et permet de partager les ressources. Elle repose sur des fonctions de désactivation et de réactivation.

Un objet d'ingénierie de base d'une grappe peut se trouver en interaction avec d'autres objets. Lors de la désactivation de la grappe, l'objet d'ingénierie de base est sauvegardé dans un gabarit de grappe, mais ses activités sont gelées et il ne peut plus continuer ses interactions avec d'autres objets. La transparence à la persistance cache la désactivation et la réactivation des gabarits de grappes, en sorte que les objets d'ingénierie de base paraissent toujours disponibles pour des interactions. Cette propriété implique que l'objet possède une durée de vie propre indépendante de l'environnement sur lequel il repose.

### 8.9.6      **Transparence à la relocalisation**

La transparence à la relocalisation cache à un objet de traitement le fait que les interfaces auxquelles il est lié ont changé de position. Cette propriété implique la capacité de rétablir la liaison si nécessaire. La transparence à la relocalisation repose sur la fonction de relocalisation.

### 8.9.7      **Transparence à la duplication**

La transparence à la duplication masque l'emploi d'un groupe d'objets de traitement pour la mise en œuvre d'une interface de traitement unique. Elle repose sur les fonctions de duplication et de relocalisation.

### 8.9.8      **Transparence aux transactions**

La coordination des transactions demande que l'on puisse ordonnancer, surveiller et récupérer après erreur les actions qui participent à ces transactions. Il faut à ces fins que se produisent des interactions entre les objets de traitement impliqués dans l'exécution des actions qui participent aux transactions et les objets de traitement qui exécutent la fonction de transaction. Il n'est pas possible en général de coordonner les actions participantes en se contentant de configurer les objets d'ingénierie dans les canaux de manière à capter ces actions participantes au passage. Des actions internes comme le démarrage ou la terminaison d'une transaction, en particulier, ne sont pas détectables par surveillance des interactions entre objets de traitement.

La surveillance et le gouvernement des transactions exigeront donc la présence d'interactions visibles au niveau traitement entre, d'une part, des objets de traitement qui représentent les objets d'information ou les schémas dynamiques (c'est-à-dire les objets d'application) et, d'autre part, des objets de traitement qui fournissent la fonction de transaction. Mais le traitement transactionnel est d'une complexité extrême et il n'est pas souhaitable de compliquer la spécification des fonctions exercées par l'application par l'adjonction des interactions complexes qu'exige la gestion des transactions.

La transparence aux transactions est la fourniture d'un processus automatique pour affiner une spécification de traitement dénuée de gestion de transaction en une spécification de traitement munie de gestion de transaction.

La nature de l'affinement et le niveau d'implication exigé du prescripteur dans le processus d'affinement dépendront du mécanisme transactionnel choisi. Il y aura normalement besoin d'interfaces de traitement additionnelles, qui devront être liées par le moyen d'interactions opérant sur ces interfaces aux objets de traitement impliqués dans la fonction de transaction, afin de pouvoir coordonner l'ordonnancement, la surveillance et la récupération sur erreur des actions qui participent à la fonction de transaction. Il y aura lieu de donner plus d'ampleur au comportement des objets de traitement pour y inclure des actions de reprise. Il faudra peut-être remplacer ou amplifier les interfaces qui donnent accès à leurs fonctions normales.

## 9            **Evaluation de la conformité**

### 9.1        **Evaluation de la conformité et processus de développement**

Le développement d'un produit part de l'appréhension initiale d'un ou de plusieurs besoins qu'exige un système ODP pour finir par la fourniture d'un exemplaire de système ODP qui réponde à ces besoins. Il est possible que le processus de développement implique la production de plusieurs spécifications. Toute spécification est susceptible d'entraîner la création de plusieurs spécifications subséquentes par application de démarches (ou "transformations" ) dont il existe de nombreux types, parmi lesquels on trouve:

- la traduction;
- l'affinement.

Les spécifications s'expriment dans un langage naturel ou formel. La traduction produit, dans un langage généralement différent, une spécification qui conserve la même signification. Par contre, l'affinement produit une spécification comportant de nouveaux détails qui permettent de définir plus précisément le produit. Si toute spécification se caractérise par l'ensemble des produits qu'elle est susceptible de spécifier, la traduction ne change pas cet ensemble mais le raffinement conduit à un sous-ensemble.

Après installation et mise en état, un exemplaire de système ODP entre dans une phase d'utilisation opérationnelle au cours de laquelle il est supposé répondre aux besoins exprimés dans le document dans lequel sont spécifiées les exigences qui le concernent. La capacité du produit à y réussir dépend d'un certain nombre de pratiques qui entrent en jeu durant chacune des phases du développement. Il existe des normes de "qualité" qui décrivent des ensembles cohérents de pratiques de ce genre ainsi que les organisations en termes de personnes, de documentation et de cycle opératoire auxquelles elles s'appliquent. Des mesures de qualité sont normalement effectuées à chaque phase et des modifications interviennent si la qualité est jugée insuffisante. L'évaluation de la conformité constitue l'une de ces mesures de qualité, qui intervient d'habitude au cours de la phase pendant laquelle la spécification de réalisation est mise en œuvre. Elle peut néanmoins avoir une utilité ou porter à conséquence dans d'autres phases.

## **9.2 Evaluation de la conformité: relations pertinentes**

Les relations entre spécifications et mises en œuvre effectives qui sont pertinentes pour la conformité se répartissent en deux groupes:

- i) relations entre spécifications et mises en œuvre effectives (conformité);
- ii) relations entre spécifications seules (compatibilité, affinement, cohérence et validité interne).

La conformité est une relation entre une spécification et une mise en œuvre effective, par exemple dans un produit. Elle est satisfaite lorsque certaines exigences particulières de la spécification, appelées exigences de conformité, sont observées par la réalisation. L'évaluation de la conformité est le processus d'étude de cette relation.

La compatibilité est une relation entre deux spécifications A et B: il y a compatibilité de la spécification B avec la spécification A lorsque A exprime des exigences que vérifie B.

Au cours du processus de développement, la conformité d'une mise en œuvre effective n'est pas toujours évaluée par rapport à la spécification de réalisation, au "niveau le plus bas". Il est possible de se servir d'une spécification de "niveau plus élevé" celle, par exemple, dont l'affinement a conduit à la spécification de réalisation. Dans tous les cas, la correspondance entre spécifications consécutives revêt une grande importance. Une de ces correspondances est déterminée par les règles d'affinement.

Deux spécifications sont reliées par une relation d'affinement lorsque l'une est un affinement de l'autre et que tous les produits capables de conformité à la spécification affinée sont aussi conformes à la spécification qui est à l'origine de l'affinement. On s'assure ainsi que les contraintes de la spécification la plus générale sont présentes dans la spécification affinée.

L'appareillage entre spécifications n'a pas toujours pour origine qu'elles découlent du même jeu d'exigences. Elles ont quelquefois été élaborées à partir de deux ou de plusieurs points de vue fort différents, ce qui est le cas, par exemple, de spécifications en langages de points de vue distincts. En ce cas, la cohérence entre spécifications peut poser problème: il importe que les exigences exprimées par une des spécifications ne contredisent pas celles qu'exprime l'une des autres. La cohérence est une relation entre deux spécifications qui est satisfaite s'il peut exister au moins un exemplaire d'un produit qui soit conforme aux deux spécifications.

Une spécification est valide s'il n'existe pas de conflits entre ses propriétés et les propriétés implicites qu'on en attend (on peut attendre par exemple d'une spécification de protocole qu'elle ne conduise pas à des situations de blocage) et qu'il existe au moins un exemplaire d'un produit qui soit capable de lui être conforme. Autrement dit, elle est dénuée de contradictions internes.

L'évaluation de la conformité est l'étude de ces relations. Elle s'effectue soit par des essais, dans le cas de la conformité, soit par vérification des spécifications, dans les cas de compatibilité, d'affinement, de cohérence et de validité interne.

## **9.3 Points de conformité et concepts apparentés**

Lorsque la conformité d'une réalisation d'une spécification ODP est évaluée par le moyen d'essais de conformité, on analyse son comportement en l'excitant pour observer les événements résultants en des points d'interaction bien précisés. Ces points sont appelés "points de conformité". Ils sont normalement choisis au sein d'une collection de points de même nature dont la localisation est spécifiée dans l'architecture du modèle de référence ODP. Ces points de conformité potentiels sont appelés points de référence.

Afin de respecter la compatibilité avec le modèle de référence ODP, les spécifications ODP normalisées ont l'obligation d'inclure une déclaration de conformité qui doit, entre autres choses, énoncer les points de référence à utiliser au cours des essais de conformité. L'idée est que tout point de conformité que mentionne une spécification doit être l'un des points de référence définis dans l'architecture.

Les concepts de points de conformité, de points de référence et de points d'interaction ont été ébauchés ci-dessus. L'ISO/CEI 9646, qui spécifie un cadre général et une méthodologie pour les essais de conformité de l'OSI, définit la notion de point de contrôle et d'observation (PCO, *point of control and observation*), qui n'est pas le point où la spécification exige que le comportement soit conforme à la norme, et donc pas un point de conformité, mais le point où l'on régit et d'où l'on observe le comportement à un point de conformité.

On spécifie un point de conformité dans celles des spécifications ODP par rapport auxquelles on peut s'attendre à voir revendiquer la conformité. On spécifie un point de contrôle et d'observation lorsqu'on décrit une méthode particulière d'essai d'une réalisation. Des méthodes différentes sont susceptibles d'impliquer l'emploi d'ensembles différents de PCO.

Il est possible de décrire le système de traitement utilisé pour exécuter les essais en utilisant la nomenclature ODP. Il est à noter que les PCO de l'application soumise aux essais sont essentiellement les points de conformité du système qui la teste. Les points de référence du système de test sont des PCO potentiels dont la prise en considération pour la définition d'une méthode d'essais peut avoir été spécifiée au préalable.

Les fondements et l'architecture que spécifient respectivement la Rec. UIT-T X.902 | ISO/CEI 10746-2 et la Rec. UIT-T X.903 | ISO/CEI 10746-3 ne définissent pas séparément de PCO ni de PCO potentiels. On peut utiliser à leur place les notions de points de conformité et de points de référence du système de test. Localiser des PCO potentiels aux seuls points de référence assurerait que les systèmes de test sont eux-mêmes des systèmes ODP, ce qui limiterait le champ des méthodes d'essais susceptibles de normalisation. Quand l'usage de l'ODP se répandra, comme il n'y aura pratiquement plus de distinction entre PCO potentiels et points de référence, le terme PCO deviendra probablement superflu.

## 9.4 Spécifications de conformité ODP

### 9.4.1 Niveau d'abstraction

Il se peut que les testeurs de conformité aient besoin d'informations complémentaires lors des essais portant sur une implémentation de spécification ODP. Ce sont les informations complémentaires nécessaires aux essais de conformité d'une implémentation (IXIT, *implementation extra information for testing*). Elles comprennent toutes les informations nécessaires pour relier les concepts présentés dans les déclarations de conformité d'une implémentation (ICS, *implementation conformance statement*) à leur implémentation dans des réalisations.

Le niveau d'abstraction auquel est spécifié un point de conformité a des conséquences sur le travail qu'il faut accomplir pour fournir les IXIT, sur les contraintes exercées sur le processus de développement et sur le degré de liberté offert à l'implémentation:

- il y a besoin de plus d'informations complémentaires sur le passage de la spécification à la réalisation lorsque l'on part d'un plus haut niveau d'abstraction. Comme la fourniture de ces informations et le processus d'implémentation impliquent tous deux que l'on relie la terminologie de la spécification aux objets façonnés dans l'implémentation, l'expression des IXIT peut exiger une quantité de travail proportionnelle à celle que requiert l'ensemble du processus de développement. Dans le cas de points de conformité définis à un niveau particulièrement élevé d'abstraction, on peut déboucher là sur un coût déraisonnable qui peut rendre peu séduisants les essais de conformité;
- il faut noter par ailleurs que tous les PCO doivent être matérialisés de manière explicite et accessible dans une implémentation, ce qui n'est pas nécessairement le cas pour d'autres éléments de la spécification.

La spécification des points de conformité, auxquels il faut allouer des PCO, impose donc sur le processus de développement des contraintes qui croissent avec le nombre de points de conformité définis. Moins la spécification d'un point de conformité est reliée à la réalisation, c'est-à-dire plus est élevé son niveau d'abstraction, plus augmentent les inconvénients qui résultent de la nécessité d'offrir une représentation explicite dans l'implémentation effective;

- utiliser pour la spécification d'un point de conformité un niveau d'abstraction moins élevé entraîne que moins de détails ont été traités d'une manière abstraite par rapport à la réalisation et que par conséquent la spécification est moins indépendante de la réalisation. La plus grande précision des détails implique aussi que le processus d'évaluation de la conformité risque de demander plus de travail puisqu'il y aura plus de détails à vérifier, bien qu'il y ait besoin de moins d'interprétation pour fournir l'IXIT.

#### 9.4.2 Usage de plusieurs points de référence

Il peut s'avérer nécessaire de définir dans une spécification plusieurs points de conformité plutôt qu'un, pour des raisons qui ne tiennent pas seulement à l'augmentation du nombre des aspects prescriptifs de la spécification.

La spécification d'un objet peut comprendre la définition de plusieurs interfaces de différents types entre lui et son environnement. Il peut exister, par exemple, des interfaces avec des personnes, avec d'autres objets, avec des mécanismes de communication ou avec des mécanismes de mise en mémoire d'information. Chacune d'entre elles peut se voir caractériser par des points de référence appartenant à différentes classes, physique, de programmation, d'interfonctionnement ou d'échange, qui emportent des conséquences différentes sur les essais de conformité. Il faudra donc définir des points de conformité différents dès lors que seront présentes des interfaces de ces différents types.

Même si l'on ne considère qu'un seul type d'interface, il peut avoir été prévu que la réalisation d'un objet serait accessible à des points physiquement ou logiquement distincts. L'association de points de conformité distincts à chacun d'eux constitue une reconnaissance de ces détails d'implémentation, ce qui abaisse le niveau d'abstraction auquel sont spécifiés les points de conformité puisque sont alors prescrits plus de détails de réalisation. Les effets de cet abaissement ont été indiqués dans leurs grandes lignes au 9.4.1. Enoncer des points de conformité distincts donne un tour prescriptif et non plus seulement descriptif à "l'intention" de disposer de plusieurs points d'accès et assure par là que cette intention sera suivie d'effet.

Lorsque la spécification d'un objet prend la forme d'un certain nombre de composants présentant des interfaces de liaison entre eux, il n'est pas toujours clair si les détails internes ne servent qu'à prescrire le comportement externe de l'objet par implication, ou s'ils emportent un caractère prescriptif qui oblige à construire l'objet de la manière décrite. Quand les composants sont tenus pour des détails normatifs, il faut définir des points de conformité additionnels, par exemple à chaque interface entre composants.

### 9.5 Conséquences des langages de points de vue sur la conformité

L'emploi de langages de points de vue ODP pour les spécifications ODP entraîne un certain nombre de conséquences:

- les testeurs de conformité sont censés évaluer les effets des spécifications selon les termes de la spécification d'ingénierie de la réalisation en cours d'essais. Pour cette raison, entre autres facteurs, il faudra que le client du laboratoire d'essais fournisse des informations complémentaires nécessaires aux essais de conformité, soit:
  - i) des IXIT et des ICS pour relier l'implémentation des concepts et des structures de la spécification d'entreprise d'une réalisation à l'implémentation de sa spécification d'ingénierie;
  - ii) des IXIT et des ICS pour relier l'implémentation des concepts et des structures de la spécification d'information d'une réalisation à l'implémentation de sa spécification d'ingénierie;
  - iii) des IXIT et des ICS pour relier l'implémentation des concepts et des structures de la spécification de traitement d'une réalisation à l'implémentation de sa spécification d'ingénierie;
  - iv) des IXIT et des ICS pour relier l'implémentation des concepts et des structures de la spécification d'ingénierie à l'implémentation des choix effectués dans sa spécification de technologie. Ces informations font partie de la spécification de technologie;
- chaque langage de point de vue permet de spécifier séparément différents types d'exigences, en sorte qu'on puisse, par exemple, traiter séparément des objectifs d'affaires, de la conception du système et de l'emploi des solutions techniques. Elaborer en parallèle des spécifications distinctes est par conséquent possible;
- vérifier la production en parallèle de spécifications implique que soit évaluée de temps à autres au cours du processus d'élaboration la cohérence entre les spécifications.

Comme les spécifications fournies sous les différents points de vue peuvent être indépendantes les unes des autres et peuvent, par conséquent, ne pas être apparentées par une relation d'affinement, la vérification d'affinement ne suffit pas à une vérification complète tout au long de l'avancement en parallèle des différentes spécifications. Il est cependant de toute évidence souhaitable qu'une vérification de cette nature soit possible. C'est ce qu'apporte la vérification de cohérence décrite au 8.7. Les objets que décrivent les spécifications de points de vue appartiennent souvent au même univers de discours: un objet d'information d'une spécification d'information peut par exemple apparaître comme argument d'une fonction dans l'interface d'un objet de traitement, les points de conformité de la spécification d'entreprise peuvent réapparaître sous forme de points de conformité dans les spécifications de traitement et d'ingénierie, etc.

## 9.6 Activités d'évaluation de la conformité

Les activités à entreprendre au cours du processus d'évaluation de la conformité comprennent:

- vérification d'affinement, entre spécifications;
- vérification de validité interne, pour une spécification;
- vérification de cohérence, entre plusieurs spécifications;
- essai d'une réalisation ou d'une simulation, entre une implémentation effective et une spécification.

## 10 Gestion des systèmes ODP

Un système ODP se compose d'un certain nombre d'applications ODP et de services de soutien. Parmi ces derniers figurent non seulement des services qui, comme le traitement, le stockage de fichiers, l'accès des utilisateurs ou les communications, sont ceux qu'offrent les systèmes d'exploitation traditionnels des machines centrales, mais aussi les services qu'exige la répartition du système, comme les annuaires et la gestion des noms, le courtage et les services qui prennent en charge les transparences à la répartition. Il faut gérer tous ces services en même temps que les applications ODP. La nature des fonctions de gestion requises à cet effet dépendra des services et des applications concernés.

La gestion d'un système ODP sera exercée elle-même par une ou plusieurs applications de gestion travaillant en interaction avec les services du système et les applications ODP au travers d'interfaces. Le niveau de détail atteint par la gestion dépend de l'application de gestion concernée. Lorsqu'il y aura lieu, l'application de gestion sera en interaction avec les fonctions ODP au travers des fonctions de gestion qui ont été définies à leur intention.

Les applications de gestion d'un système ODP doivent refléter les relations qui existent entre les fonctions de gestion. Le fait, par exemple, qu'il puisse être nécessaire de reconfigurer le système pour en maintenir le niveau de qualité de service, démontre que les fonctions de gestion entretiennent des relations mutuelles. La tenue des données comptables fait aussi normalement partie des applications de gestion, parce que la plupart des services ont besoin de tenir une comptabilité interne pour leur propre gestion, même s'il n'est pas exigé de leurs utilisateurs qu'ils paient le service rendu.

Les fonctions de sécurité ne font pas partie de l'ensemble des fonctions de gestion, bien qu'il soit envisagé de soumettre les interactions de gestion à des directives politiques. Il est généralement tenu pour essentiel, par exemple, de faire appel à l'authentification et au contrôle d'accès pour empêcher des utilisateurs ou des gestionnaires non autorisés d'effectuer des actions de gestion sur des composants d'un service ou d'une application. Tous les services d'un environnement ODP ont donc besoin d'un service de sécurité, tout comme ils ont besoin d'un service de communication. Mais la gestion de la sécurité ne constitue pourtant pas un service de gestion d'ordre général nécessaire pour gérer d'autres services.

### 10.1 Domaines de gestion

Afin de faire face à la complexité de la gestion et aux problèmes d'échelle, tout particulièrement dans le cas des grands systèmes ODP, il est essentiel de disposer d'un cadre commun permettant de subdiviser la gestion globale. Au sein de l'environnement de l'ODP coexistent un grand nombre de vues sur la gestion et de frontières de responsabilité, qui reposent sur des critères de structuration chaque fois différents.

Les domaines de gestion donnent le moyen de spécifier avec souplesse et pragmatisme des frontières de responsabilité et d'autorité de gestion qui reflètent ces différentes vues. Un domaine détermine un ensemble d'objets, tous individuellement liés par une relation caractéristique avec un objet de contrôle du domaine. Les objets membres peuvent être des ressources, des stations de travail, des modems, des processus, etc., en fonction de l'objectif qui justifie la définition d'un domaine donné. L'objet de contrôle est muni d'un attribut d'identification des objets membres. Un objet est tenu pour objet membre du domaine si son identité est connue de l'objet de contrôle du domaine de gestion.

Les domaines de gestion permettent le gouvernement d'un ensemble d'objets gérés sous l'angle d'une politique unique, base à partir de laquelle il devient possible de faire face à la complexité des grands systèmes ODP. Ils rendent plus faciles les activités de gestion parce qu'il devient possible de modifier la politique et le sociétariat du domaine par le moyen d'interactions avec un objet unique, l'objet de contrôle du domaine de gestion, plutôt que d'obliger des gestionnaires à intervenir individuellement sur chacun des nombreux objets qui participent à l'environnement.

Les domaines de gestion ne mettent pas les objets membres sous capsule: il est permis à des objets externes d'entrer directement en interaction avec un objet du domaine. Les domaines de gestion persistent même s'il peut leur arriver à certains moments de ne contenir aucun objet car il doit être possible de créer un domaine de gestion vide dans lequel des objets seront introduits à une date ultérieure.

## 10.2 Politique de gestion

En termes d'entreprise, la relation caractéristique d'un domaine incarne la politique associée au domaine. Les domaines de gestion donnent ainsi les moyens de spécifier la politique de gestion qui s'applique à un groupe d'objets gérés, plutôt que d'avoir à le faire pour chaque objet individuellement. Les objectifs globaux de gestion et les contraintes externes d'ordre légal, comme les lois nationales sur la protection des données, réglementaire ou politique à haut niveau, forment deux des aspects de la politique afférente à un domaine de gestion. Ces exemples montrent qu'il peut s'avérer difficile de spécifier formellement certaines politiques.

Des contraintes internes restreignent les opérations qu'il est possible d'exécuter sur les objets d'un domaine de gestion. Elles peuvent s'exprimer de manière déclarative en termes d'obligations portant sur les membres potentiels du domaine.

Un aspect d'importance pour une politique de gestion est la spécification des opérations de gestion que les gestionnaires sont autorisés à exécuter sur les objets qu'ils gèrent. Une règle d'accès est une relation d'autorité qui spécifie l'ensemble des interactions permises entre un domaine de gestionnaires et un domaine d'objets gérés. Tous les membres du domaine "SysProgrammers" sont par exemple autorisés à démarrer et à arrêter les objets du domaine "DepartmentServices". Les interactions autorisées peuvent constituer un sous-ensemble des interactions de gestion définies par les interfaces des objets du domaine.

## 10.3 Modélisation des structures de gestion

Il est possible d'utiliser des relations de domaines pour modéliser les structures de gestion. Deux domaines de gestion sont dits disjoints s'ils ne possèdent aucun objet en commun. Deux domaines de gestion se chevauchent s'il existe des objets qui sont membres des deux domaines. Un exemple en est donné par le partage, entre les centres de gestion de deux domaines, de la gestion d'un portail placé entre ces domaines. On obtient ce résultat en référençant l'objet dans les deux domaines de gestion.

Un chevauchement implicite risque de se produire entre deux domaines de gestion qui contiennent des objets de types différents mais qui font référence à la même entité du monde réel. Un exemple en est donné par le cas de domaines de planification et d'entretien où la mise hors service d'une station de travail dans le domaine d'entretien la rend indisponible dans le domaine de planification. Des chevauchements implicites ont des chances de se produire lorsque la répartition de la gestion entre domaines de gestion différents s'effectue sur des bases fonctionnelles.

# 11 L'emploi des normes dans les systèmes ODP

Le présent article a pour objectif d'illustrer au moyen d'un exemple simple l'emploi dans un système ODP de normes autres que les *normes ODP* spécifiques et de montrer en particulier où, dans la spécification d'un système ODP, est défini cet emploi. A cette fin, l'exemple brosse à grands traits le contenu de chacune des spécifications de points de vue d'un système exemple et, pour chacune des spécifications de points de vue, discute des types de normes qui conviennent pour répondre aux exigences qu'elle exprime.

La Figure 16 illustre le système exemple. Il s'agit d'un système serveur avec opérateur local, relié, par un moyen de télécommunication, à un système client formé d'une station de travail assortie de deux opérateurs locaux. La description du système concerne une application de la station de travail qui permet à ses opérateurs locaux d'accéder à une application de fourniture d'images du système serveur. On suppose que tant le serveur que la station de travail offrent d'autres services et que les organisations auxquelles ils appartiennent sont distinctes, ce qui les soumet à des politiques administratives distinctes.

Les descriptions de points de vue déterminent les exigences portant sur les spécifications communes. Celles-ci correspondront en général à des normes de différents niveaux.

## 11.1 Point de vue entreprise

### 11.1.1 Spécification d'entreprise

La Figure 17 illustre le modèle par objets relatif à la spécification d'entreprise du système représenté par la Figure 16. Dans ce modèle:

- utilisateur d'images *n* est un objet d'entreprise qui représente une personne et qui joue le rôle "Utilisateur d'images *n*";
- fournisseur d'images est un objet d'entreprise qui représente un système d'information IT et qui joue le rôle de "Fournisseur d'images";

- fournisseur d'accès aux images est un objet d'entreprise qui représente un système d'information IT et qui joue le rôle de "Fournisseur d'accès aux images".

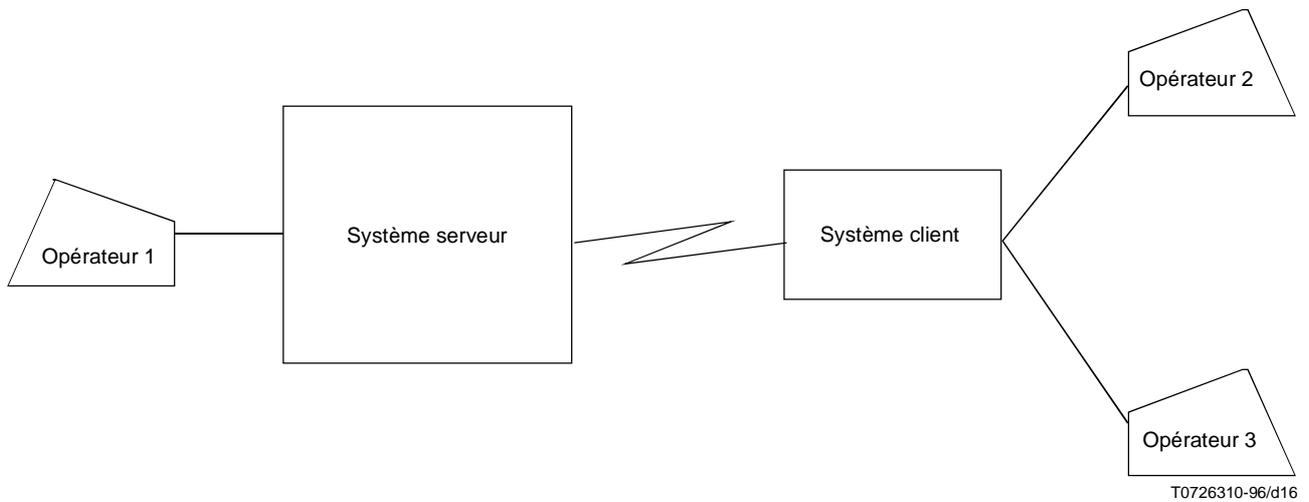


Figure 16 – Configuration du système exemple

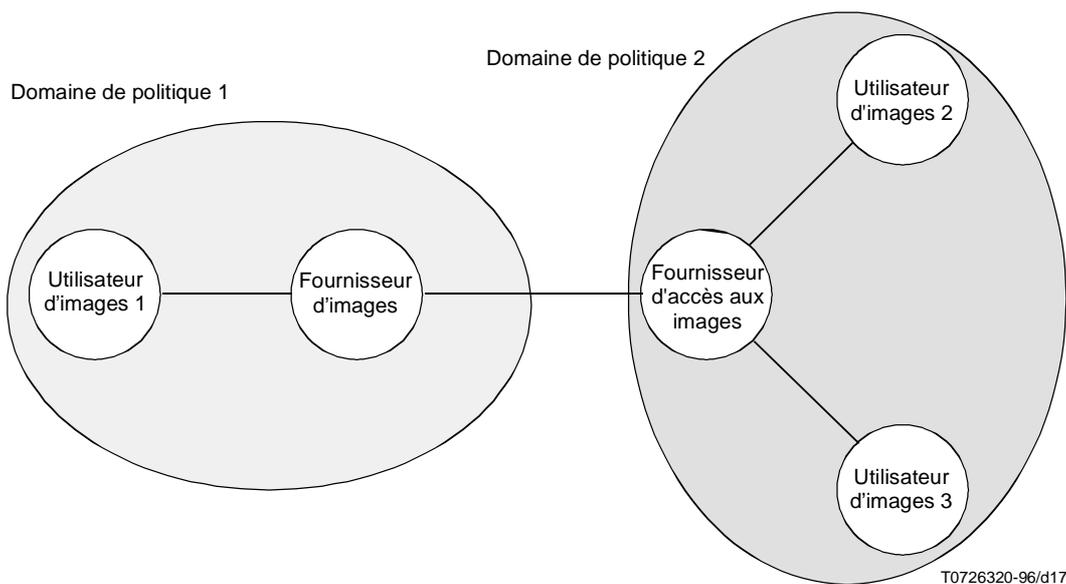


Figure 17 – Point de vue entreprise

Les objectifs de l'entreprise représentée par cette configuration d'objets d'entreprise régissent la nature des comportements associés aux rôles tenus par ces objets. Pour atteindre ces objectifs, il pourrait suffire par exemple de demander l'affichage d'images fixes chez les Utilisateurs d'images 2 et 3; ou bien il pourrait être exigé que Fournisseur d'accès aux images donne les moyens de manipuler les informations d'images, d'en ajouter et de mettre à jour Fournisseur d'images.

La spécification d'entreprise définit ainsi les objectifs de la configuration d'objets d'entreprise et par suite:

- les besoins en informations d'images d'Utilisateurs d'images 1, 2 et 3;
- les interactions d'Utilisateur d'images 1 avec Fournisseur d'images;

- les interactions d'Utilisateur d'images 2 et d'Utilisateur d'images 3 avec Fournisseur d'accès aux images;
- les interactions entre Fournisseur d'accès aux images et Fournisseur d'images;
- les politiques, y compris en matière de sécurité, qui gouvernent les interactions entre les objets d'entreprise;
- les exigences de qualité de service relatives à Utilisateurs d'images 1, 2 et 3.

### 11.1.2 Application des normes

Dans la description d'entreprise, on pourrait élaborer des spécifications communes pour des caractéristiques particulières à un domaine industriel ou à des fonctions commerciales, comme des *politiques de sécurité*, qu'il y a lieu de soutenir en termes de contrôle d'accès et de niveau de protection. Ces normes se rapporteraient au domaine commercial ou applicatif concerné.

## 11.2 Point de vue information

### 11.2.1 Spécification d'information

Dans le point de vue information, le système est visible en termes:

- des classes d'objets d'information impliquées dans l'application;
- des activités d'information (transition d'état des objets d'information) qui constituent l'application;
- des contraintes portant sur les transitions d'état qui peuvent se produire dans les objets d'information.

La spécification d'information contient:

- La spécification des classes d'objets elles-mêmes, comme, par exemple:
  - des gabarits d'objets d'information d'images qui déterminent les classes d'objets d'information d'images qui sont disponibles;
  - des gabarits d'objets d'affichage d'images qui déterminent les classes d'objets d'information d'images qui peuvent être disponibles. Un gabarit d'objet d'affichage d'image définit une composition d'objets d'information d'images;
  - un gabarit d'objet d'information de requête: un objet d'information de requête comprend les informations, dont des informations de contrôle pour, par exemple, la sécurité, qui sont nécessaires pour demander un affichage;
  - un objet d'*information de contrôle d'accès* qui comporte les informations nécessaires pour valider un objet d'information de requête.
- Les contraintes qui portent sur les configurations d'objets d'information comme, par exemple:
  - l'ensemble des classes de gabarit des objets d'information d'images dont dispose Fournisseur d'accès aux images est le même que celui dont dispose Fournisseur d'images;
  - l'ensemble des classes de gabarit d'objets d'affichage d'images dont dispose Utilisateur d'images 1 est un sous-ensemble de l'ensemble dont dispose Fournisseur d'images;
  - l'ensemble des classes de gabarit des objets d'affichage d'images dont disposent Utilisateurs d'images 2 et 3 sont des sous-ensembles de l'ensemble dont dispose Fournisseur d'accès aux images.

Pour le système en question, un exemple simplifié d'activité d'information pourrait être:

- création de l'objet d'information de requête dans l'état non validé;
- interaction entre l'objet d'information de requête et l'objet d'information de contrôle d'accès avec passage du statut de l'objet d'information de requête à l'état validé;
- interaction entre l'objet d'information de requête et les objets d'information d'images avec création d'objets d'affichage d'images.

### 11.2.2 Application des normes

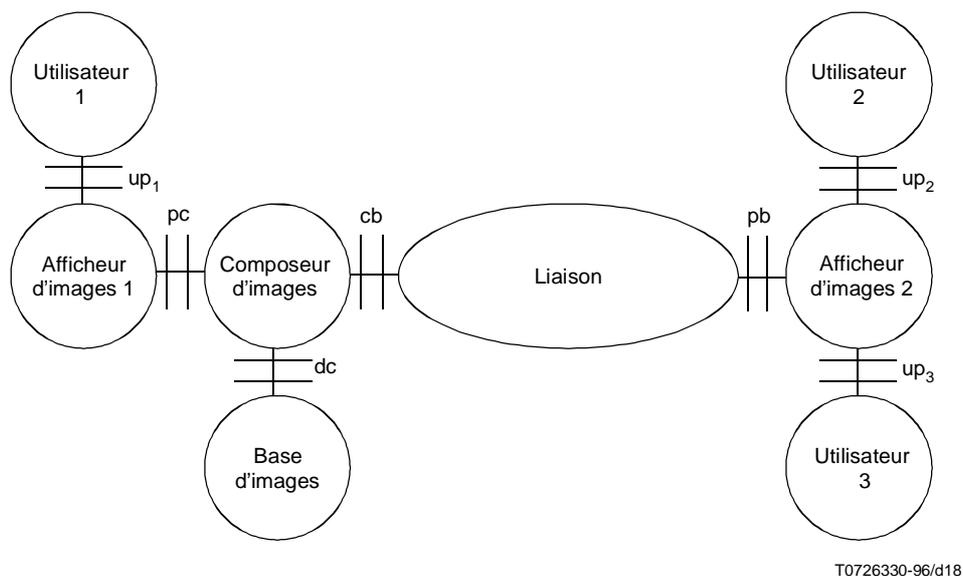
Dans la spécification d'information, on pourrait élaborer des spécifications communes pour les objets d'information. Ces normes se rapporteraient au domaine commercial ou applicatif concerné.

## 11.3 Point de vue traitement

### 11.3.1 Spécification de traitement

La Figure 18 est une illustration du modèle par objets pour la spécification de traitement du système représenté par la Figure 16. Sa configuration comporte:

- les objets de traitement Utilisateurs 1, 2 et 3, Afficheurs d'images 1 et 2, Compositeur d'images et Base d'images;
- des liaisons primitives, pour lesquelles il n'y a pas de déclaration explicite de contrat d'environnement, entre les interfaces suivantes:
  - Utilisateur 1 et Afficheur d'images 1 ( $up_1$ );
  - Utilisateurs 2 et 3 et Afficheur d'images 2 ( $up_2$  et  $up_3$ );
  - Afficheur d'images 1 et Compositeur d'images ( $pc$ );
  - Base d'images et Compositeur d'images ( $dc$ );
- une liaison composite qui met en jeu un objet de liaison (liaison) entre les interfaces d'Afficheur d'images 2 et de Compositeur d'images, avec déclaration des contrats d'environnement;
- les activités spécifiées pour la configuration d'objets de traitement qui met en œuvre les exigences spécifiées dans les spécifications d'entreprise et d'information. Les interfaces sont des points de référence de conformité au sens de l'ODP.



T0726330-96/d18

**Figure 18 – Point de vue traitement**

Du point de vue traitement, le système est visible dans les termes suivants:

- les spécifications des objets de traitement, sous la forme de spécifications abstraites des opérations et des comportements qu'ils admettent sur leurs interfaces;
- une spécification du gabarit de l'objet de liaison, comportant:
  - une spécification abstraite des opérations d'interfaces impliquées;
  - une spécification de contrats d'environnement en cohérence avec les exigences de qualité de service d'entreprise;

- une spécification des types de données abstraits qui correspondent aux objets d'information déterminés dans le point de vue information;
- les spécifications des activités qui peuvent se produire pour la prise en charge de l'application.

### 11.3.2 Application des normes

Dans la description de traitement, on pourrait élaborer des spécifications communes pour les interfaces indiquées dans la Figure 18. Elles seraient définies en termes d'opérations abstraites et de types de données abstraits correspondant aux objets d'information de la description d'information. Les normes concernant les opérations abstraites pourraient être communes à plusieurs domaines d'application, celles concernant les types de données abstraits seraient définies pour le domaine d'application concerné. Une description fondée sur OMG/CORBA et exprimée par exemple en IDL d'OMG/CORBA, est une description de traitement.

## 11.4 Point de vue ingénierie

### 11.4.1 Spécification d'ingénierie

La Figure 19 est une illustration partielle du modèle par objets pour la spécification d'ingénierie du système représenté par la Figure 16. Dans ce modèle:

- l'objet d'ingénierie de base Utilisateur 1 représente Opérateur 1;
- les objets d'ingénierie de base Afficheur d'images, Compositeur d'images et Base d'images correspondent aux objets de traitement Afficheur d'images, Compositeur d'images et Base d'images;
- les objets Talon, Lieur et Protocole constituent une partie du canal qui correspond à l'objet de liaison de la description de traitement;
- nœud correspond au système serveur de la Figure 16, bien que la Figure 19 n'illustre qu'une partie de la configuration complète de capsules, de grappes, de canaux, etc., de nœud.

Du point de vue ingénierie, le système est visible dans les termes suivants:

- les spécifications de comportement des objets d'ingénierie du canal qui correspondent à l'objet de liaison du point de vue traitement, comportant:
  - les spécifications, au *point de référence d'interfonctionnement*, des protocoles qui opèrent entre objets de protocole ou entre objet de protocole et intercepteur;
  - la spécification de la représentation concrète des types de données abstraits énoncés dans la description de traitement;
  - les exigences de qualité de service;
- pour chaque point de référence, une spécification de la syntaxe dans les termes de laquelle est exprimé le comportement à ce point de référence;
- les contraintes qui, reflétant les activités de traitement spécifiées, portent sur les comportements aux points de référence.

### 11.4.2 Application des normes

Dans la description d'ingénierie, on pourrait élaborer des spécifications communes s'appliquant aux points de référence indiqués dans la Figure 19, soit:

- pour les spécifications des protocoles et des syntaxes abstraites et concrètes qui interviennent au point de référence d'interfonctionnement situé dans le canal qui correspond à l'objet de liaison de la description de traitement, par exemple:
  - pour les protocoles et les syntaxes abstraites et concrètes d'établissement et de maintien du canal, sont applicables des profils OSI (par exemple un profil d'application RDA, un profil de transport convenable et un profil de format convenable);
  - pour les syntaxes abstraites et concrètes correspondant aux types de données abstraits de la description de traitement, il faut définir des normes et des profils de formats relatifs au domaine d'application concerné;

NOTE – Les protocoles et les syntaxes concrètes visibles au point de référence d'interfonctionnement comprennent ceux qui sont nécessaires pour établir et maintenir le canal aussi bien que ceux qui correspondent aux opérations abstraites et aux types de données abstraits spécifiés pour l'objet de liaison.

- pour les spécifications d'API, comprenant les spécifications de syntaxes abstraites pour les données, qui s'appliquent aux points de référence de programmation, par exemple:
  - un profil SQL est applicable au point de référence entre Compositeur d'images et Base d'images;
  - un profil d'API correspondant au service offert par le profil RDA est applicable au point de référence entre Compositeur d'images et Talon;
  - un profil d'API pour service de fenêtrage est applicable au point de référence entre Compositeur d'images et Afficheur d'images. Il pourrait comprendre des normes graphiques;
- pour les spécifications de HCI qui s'appliquent au *point de référence physique*, une norme de GUI par exemple.

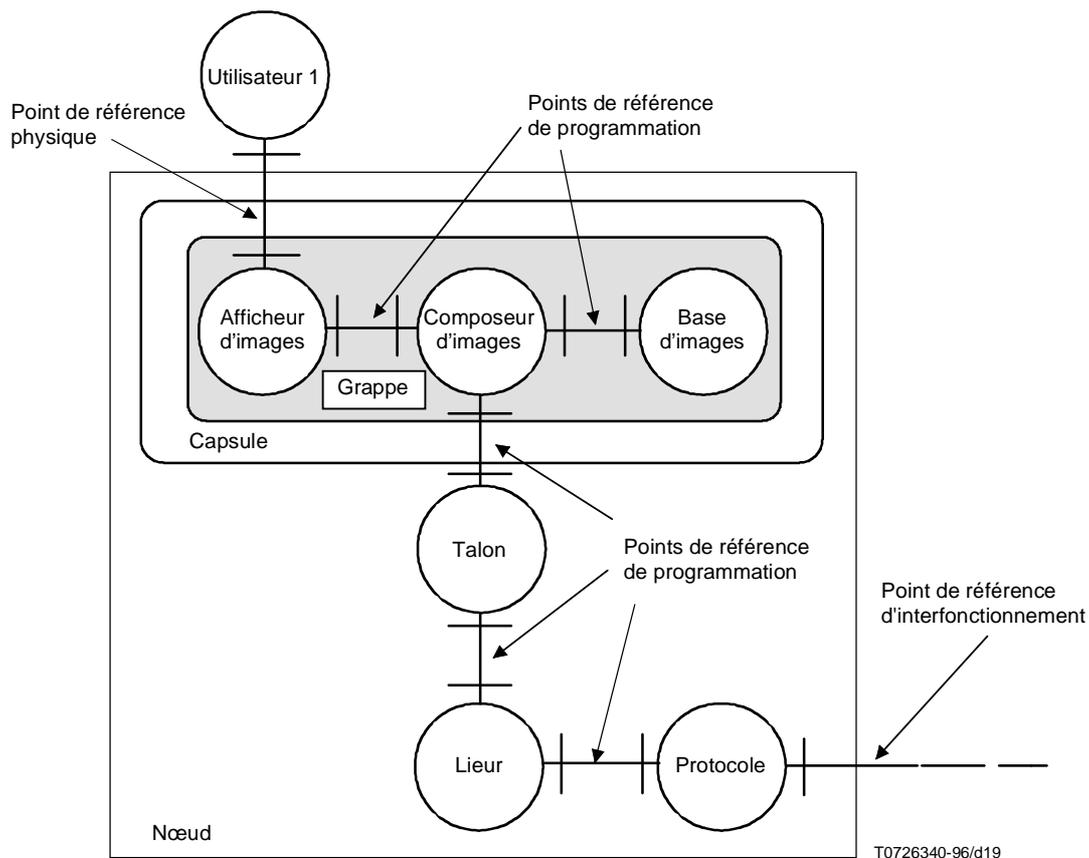


Figure 19 – Point de vue ingénierie (partiel)

## 11.5 Point de vue technologie

### 11.5.1 Spécification de technologie

Du point de vue technologie, le système est visible en termes de déclarations du fournisseur quant à la conformité de son système. Ces déclarations:

- précisent les points de conformité du système fourni qui correspondent aux points de référence de la description d'ingénierie;
- donnent les informations complémentaires nécessaires pour tester la conformité du système au comportement spécifié à ces points de conformité.

Il est à noter que le fournisseur n'est pas obligé de rendre visibles comme points de conformité tous les points de référence.

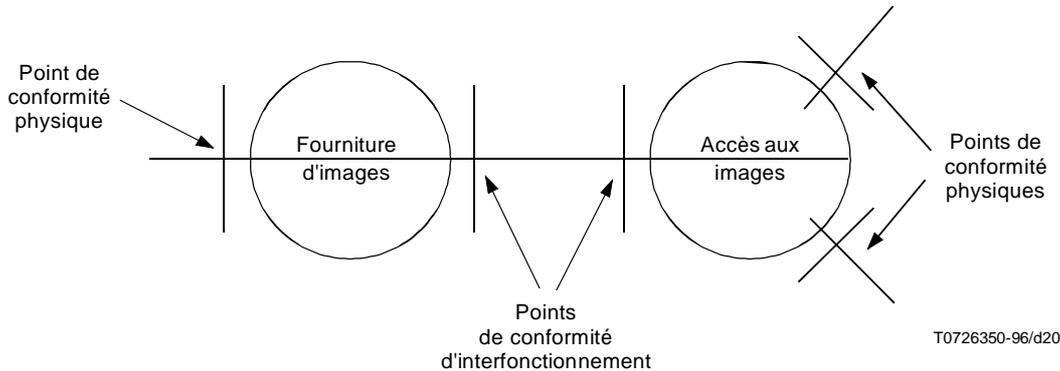


Figure 20 – Point de vue technologie (partiel)

### 11.5.2 Application des normes

Dans la description de technologie:

- l'implémentation d'un protocole OSI particulier constitue un exemple d'application des normes OSI;
- les points de conformité sont les points où le comportement du système est observable;
- les normes qui s'appliquent à ces points de conformité spécifient la syntaxe et la séquence des échanges dans les termes desquels s'exprime le comportement.

L'exigence de cohérence entre le comportement du système et les normes spécifiées pour les autres points de référence de la description d'ingénierie entraîne d'autres contraintes portant sur le comportement aux points de conformité.

## 12 Exemples de spécifications ODP

Le présent article propose plusieurs exemples d'application à la spécification de systèmes des concepts et des règles décrits dans la Rec. UIT-T X.902 | ISO/CEI 10746-2 et dans la Rec. UIT-T X.903 | ISO/CEI 10746-3. Ces exemples constituent des simplifications et des descriptions incomplètes de systèmes du monde réel, mais ils donnent un aperçu de l'emploi du cadre ODP, survolent l'application des concepts essentiels de chacun des langages de points de vue et éclairent les relations entre descriptions de points de vue.

L'exemple du 12.1 utilise les concepts et les règles du modèle de référence ODP pour concevoir un **système de téléconférence multimédia** (MMCS, *multimedia conferencing system*) sous les cinq points de vue. Le MMCS en question autorise un interfonctionnement en temps réel entre plusieurs utilisateurs utilisant de l'information multimédia, texte, vidéo et audio.

L'exemple du 12.2 spécifie un des composants appartenant au MMCS, l'**échange audiovisuel multiparties** pour systèmes répartis. Il se sert du concept de "liaison flux" pour fonder l'échange des flux audiovisuels et donne une spécification qui utilise les cinq langages de points de vue. On porte une attention particulière aux correspondances entre les cinq spécifications de points de vue afin de garantir la cohérence entre les spécifications.

L'exemple du 12.3 positionne les **concepts de gestion** dans le cadre du modèle de référence ODP.

L'exemple du 12.4 donne un aperçu général de la spécification d'une **base de données répartie**.

Les concepts dont la liste suit sont illustrés dans les différentes spécifications de points de vue de ces exemples:

**Entreprise:**

- communication et fédération;
- affectation de rôles aux objets d'entreprise;
- contrat, gabarit et politique.

**Information:**

- schéma statique;
- schéma dynamique;
- schéma d'invariant.

**Traitement:**

- spécification d'objet de traitement, y compris contrat d'environnement et comportement;
- spécification d'interfaces opération et flux;
- concept de liaison;
- référence d'interface et règles d'interaction;
- transparences.

**Ingénierie:**

- établissement de canal (objets de protocole, de liaison et talon);
- emploi des règles de structuration et des fonctions d'ingénierie pour spécifier une infrastructure compatible avec les spécifications d'entreprise, d'information et de traitement.

**Technologie:**

- choix de composants matériels et logiciels particuliers compatibles avec les spécifications des autres points de vue;
- détermination des points de conformité qu'impliquent les choix techniques.

## 12.1 Système de téléconférence multimédia

### 12.1.1 Introduction

Le système de téléconférence multimédia (MMCS) donne à plusieurs utilisateurs une possibilité d'interfonctionnement en temps réel par communication d'informations multimédia, texte, audio et vidéo. Il permet à un groupe de personnes physiquement éloignées les unes des autres de travailler en coopération sur un document multimédia et de communiquer entre elles. De nouveaux participants peuvent rejoindre une session en cours, que d'autres peuvent quitter à leur gré.

Le système donne de plus à ses utilisateurs des pouvoirs sur certains attributs du service (types d'informations souhaités, qualité d'un type d'information, etc.). Il fournit aussi un cadre pour organiser la coopération entre diverses applications.

Du point de vue de l'utilisateur, il serait désirable d'intégrer des applications comme téléconférence audiovisuelle, édition de groupe, courrier électronique. Il devrait être possible d'étendre le MMCS pour l'ouvrir.

La configuration de MMCS pourrait être constituée de stations de travail servant à présenter le texte, le son et la vidéo, selon l'exemple proposé dans la Figure 21. Des caméras, des micros et une base de données multimédia éventuellement répartie sont connectés à un réseau régional.

Pour spécifier le système correspondant à cet exemple, les règles et les concepts qui conviennent à la spécification d'un MMCS seront utilisés dans chacun des points de vue ODP. Il sort du champ de cet exemple de décrire tous les aspects d'un MMCS. Il s'agit plutôt d'illustrer comment chaque langage de point de vue participe à la spécification d'une application répartie ouverte ou d'un système réparti ouvert. On utilise la méthode d'analyse OMT [Rumbaugh 91] fondée sur le concept d'objet pour exprimer les spécifications ODP d'entreprise et d'information.

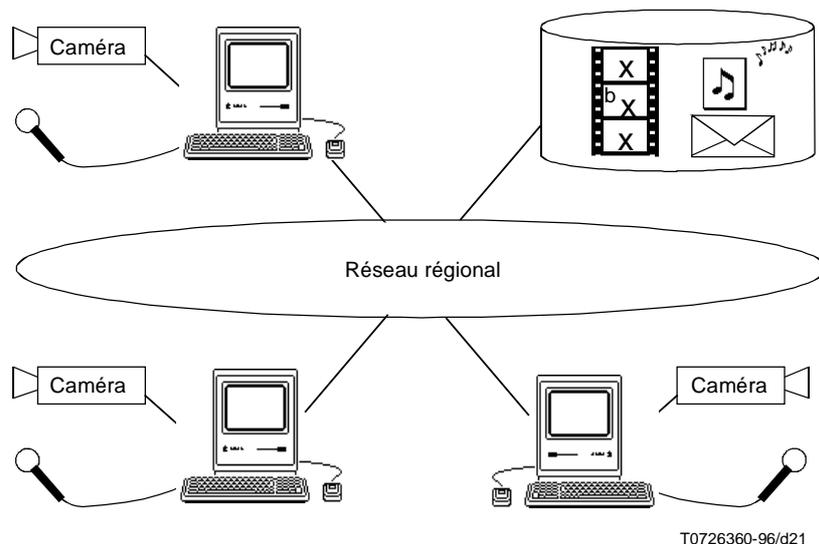


Figure 21 – Configuration d'un système de téléconférence multimédia

### 12.1.2 Spécification d'entreprise

La spécification d'entreprise décrit les objectifs, les politiques et les exigences du MMCS en question. Les exigences et les politiques du service MMCS découlent des partenaires impliqués, que l'on peut classer en fonction de leur rôle:

- utilisateur: personne ou machine qui se sert des services pour la satisfaction de certains besoins en matière de communication;
- client ou abonné: personne ou organisation qui a signé un contrat d'usage des services offerts par les prestataires de services;
- prestataire de services: organisation qui assure la gestion commerciale de services offerts aux clients selon les termes d'un accord contractuel.

Afin de structurer les exigences portant sur un service donné en fonction des rôles joués, il peut être nécessaire de spécialiser les rôles généraux introduits ci-dessus. Dans le cas du MMCS, par exemple, il est possible de distinguer deux types d'utilisateurs: le participant à la session et le meneur de session.

La question qui se pose ensuite est "que faut-il décrire pour chacun des rôles qui est impliqué dans le service?". Le modèle de référence pour l'ODP donne des indications qui recouvrent un vaste champ de politiques et de règles susceptibles d'avoir une utilité pour la description des services répartis. Les règles et les politiques générales intéressantes sont: l'usage des ressources, les règles de domaines, les règles de résolution de conflits, les règles d'organisation, les règles relatives aux affaires, les règles de transfert, les règles de sécurité, les règles de qualité de service et les règles de gestion.

La Figure 22 illustre une spécification d'entreprise simple qui emploie les concepts d'entreprise d'ODP et s'exprime dans la notation graphique OMT.

Un service de téléconférence s'étend au travers de nombreux pays différents. On peut alors imaginer une situation où les utilisateurs passeraient un contrat avec leur opérateur local de télécommunication, qui serait leur prestataire de services. Afin d'exprimer le MMCS en termes ODP, on peut utiliser les concepts de communauté et de fédération pour représenter cette situation. On peut considérer que les utilisateurs font partie d'une communauté d'utilisateurs tandis que les opérateurs de télécommunication appartiennent à une fédération. On peut analyser les relations et les interactions entre ces deux groupes et établir un accord par contrat pour ce qui concerne le MMCS.

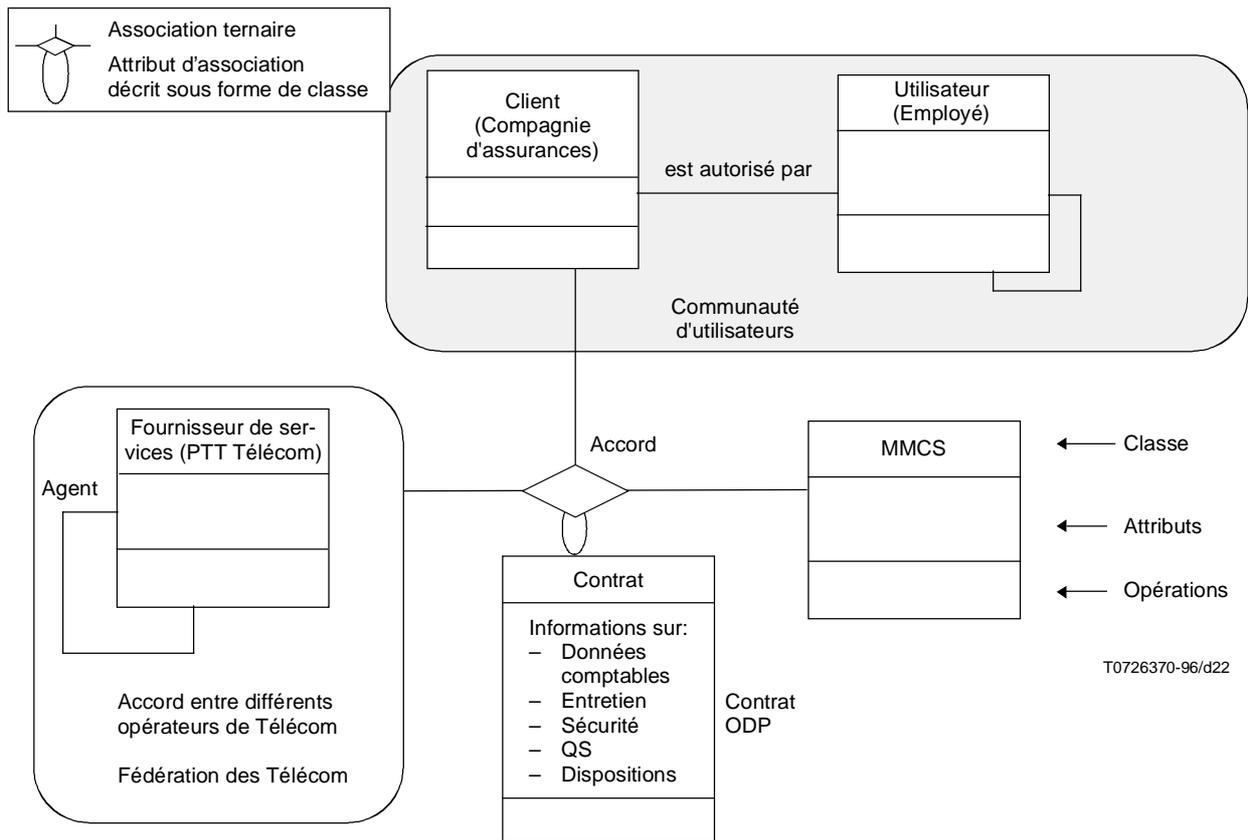


Figure 22 – Spécification d'entreprise ODP utilisant la notation graphique OMT

L'utilisateur, le prestataire de services et le MMCS peuvent tous être décrits en termes ODP sous forme d'objets d'entreprise exerçant un rôle qui leur est assigné. L'utilisateur par exemple peut avoir entrepris d'exécuter des actions en collaboration avec le MMCS par l'intermédiaire du client, et réciproquement. Ces interactions modifient les relations dites d'obligation, de permission et d'interdiction qui existent entre l'utilisateur et le MMCS. Les utilisateurs en interaction mutuelle au sein du groupe forment une communauté ODP car ils ont en commun un contrat d'obligation entre les rôles tenus par les objets d'entreprise et un ensemble d'activités.

### 12.1.3 Spécification d'information

La spécification d'information précise la sémantique et les exigences applicables au traitement de l'information du service.

Le concept de *schéma* est utilisé à cet effet. On définit un schéma local pour chaque rôle d'utilisateur et un schéma global qui représente les informations valables pour tout rôle d'utilisateur et les informations relatives au service. Il est à noter que cette spécification donne un exemple de schéma statique, mais ne propose pas d'exemples ni de schémas d'invariants ni de schémas dynamiques.

En focalisant sur les classes présentées dans la spécification d'entreprise, la Figure 23 présente la spécification d'information OMT d'un schéma statique du système au moment où une session est en cours.

Le client possède une description des paramètres de configuration du MMCS. Il dispose aussi d'informations relatives aux limites d'allocation de bande passante, une liste des utilisateurs finals enregistrés qui sont en mesure de lancer une téléconférence, une liste des options auxquelles ont droit les utilisateurs finals, etc. On peut voir ces informations client comme des attributs d'un objet Client dans un modèle OMT par objets. Un objet d'information Utilisateur se trouve connecté à l'objet d'information Client pour représenter le fait qu'il ne peut participer à une téléconférence que si dans le modèle d'entreprise l'utilisateur final correspondant y est autorisé par le client correspondant. L'utilisateur meneur est l'utilisateur qui préside la téléconférence.

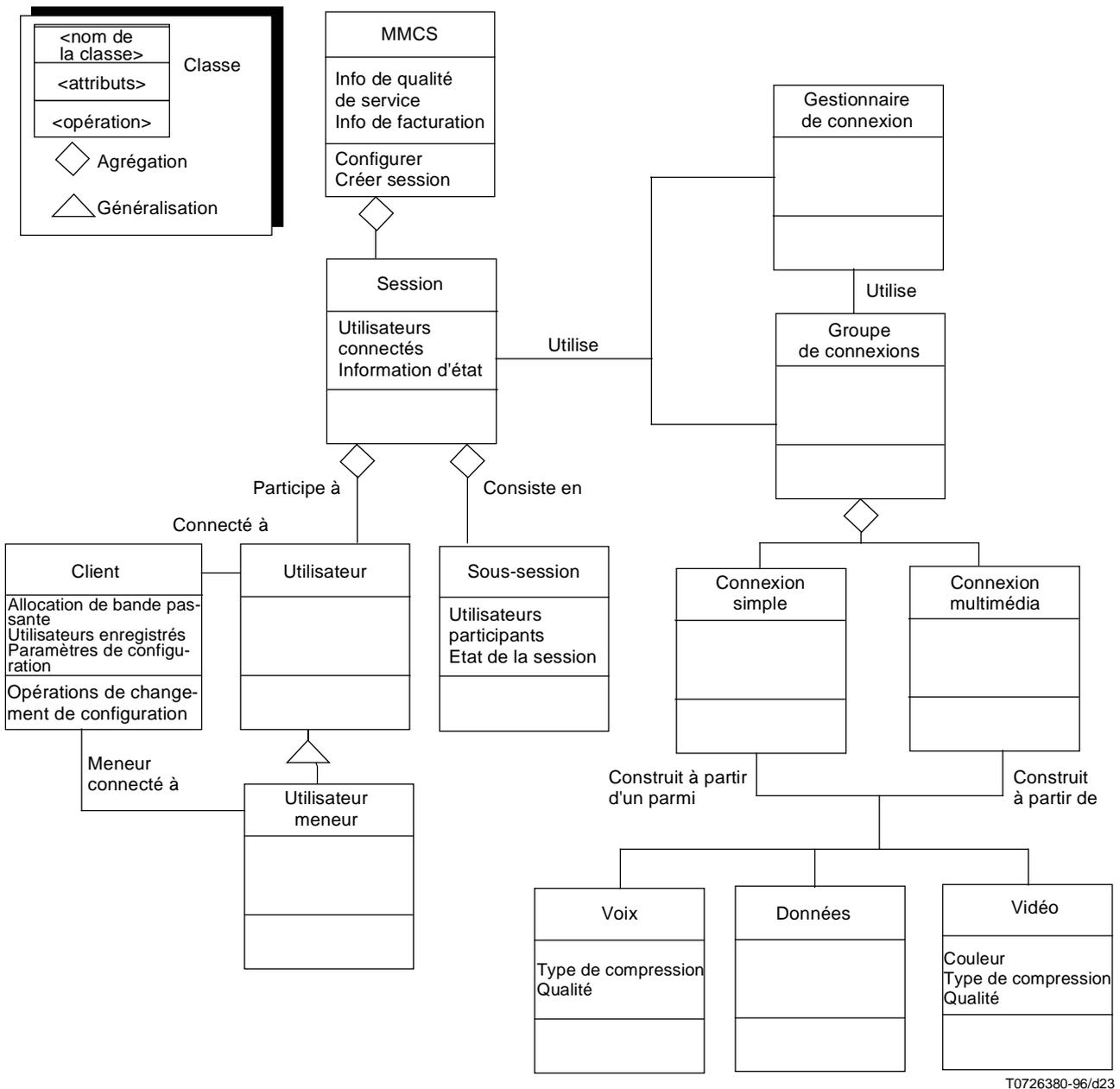


Figure 23 – Spécification d'information

### 12.1.4 Spécification de traitement

La spécification de traitement est élaborée de manière à rester cohérente avec les spécifications d'entreprise et d'information. Les objets d'information et les objets de traitement ne sont pas nécessairement en mappage un pour un. La spécification d'information est fondamentalement différente d'une spécification de traitement: les objets de traitement, en particulier, sont définis en termes d'interfaces, ce qui n'est pas le cas des objets d'information.

La correspondance entre la spécification d'information et la spécification de traitement doit toujours être spécifiée en sorte qu'il soit possible de garantir la cohérence entre ces spécifications. C'est là une tâche importante qui est impartie au concepteur du service. La Figure 24 donne une illustration des mappages dans le cas du MMCS.

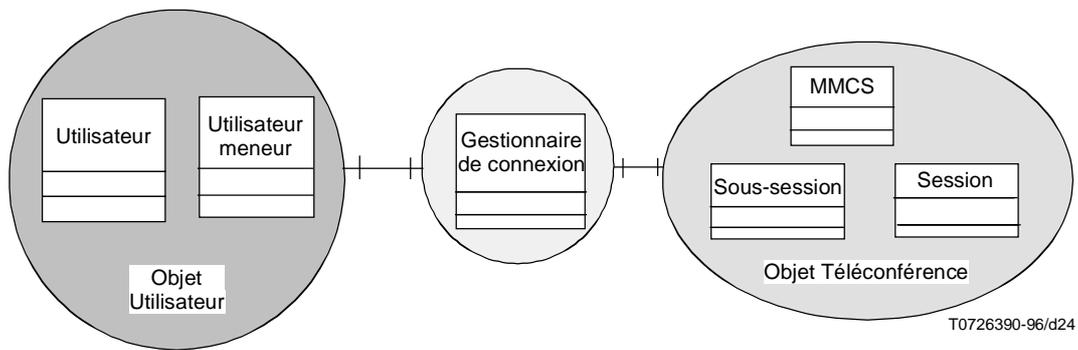


Figure 24 – Configuration de traitement du MMCS

L'analyse OMT permet de déterminer et de concevoir les objets de traitement. Ils sont déterminés par regroupement d'objets dont les fonctions sont liées. L'analyse OMT permet aussi d'effectuer un certain nombre de choix relatifs à la configuration d'ingénierie et aux infrastructures techniques.

Le concepteur de service décide du regroupement des classes en objets de traitement sans se soucier des aspects liés à la répartition:

- les objets, classes et associations OMT relatifs à l'utilisateur et à l'utilisateur meneur sont pris en compte pour la conception de l'objet de traitement Utilisateur;
- les objets, classes et associations OMT relatifs à la téléconférence, comme MMCS, session ou sous-session, sont pris en compte pour la conception de l'objet de traitement Téléconférence;
- les objets, classes et associations OMT relatifs à la connexion, comme le gestionnaire de connexion, sont pris en compte pour la conception des liaisons de traitement et aideront à choisir la configuration d'ingénierie (par exemple, décision d'employer des fonctionnalités de diffusion).

Pour chaque objet de traitement ainsi déterminé en conclusion de cette analyse, il faut définir des interfaces pour ses interactions avec d'autres objets de traitement soit, par exemple, des opérations de téléconférence ou des opérations d'émission de flux audiovisuels. Les objets et leurs interfaces sont en général graphiquement représentables comme le montre la Figure 25.

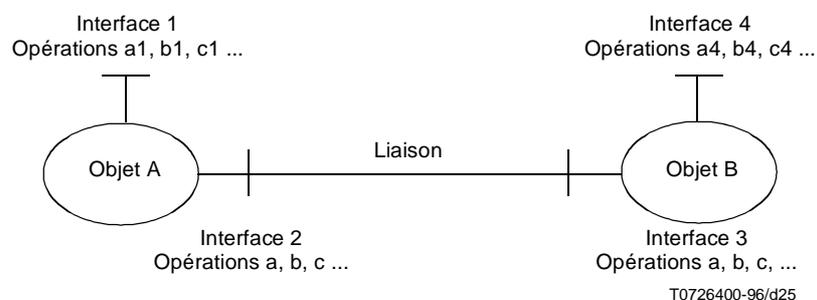


Figure 25 – Représentation de traitement des objets et des interfaces

Si l'on porte l'attention sur les interactions qui sont réalisables par l'intermédiaire d'objets de liaison, elles sont établies par des actions explicites de liaison qui donnent à l'utilisateur le moyen de spécifier la liaison requise entre les objets de traitement. Un objet de liaison qui résulte d'une liaison explicite est en mesure de prendre en charge des échanges de type opération entre les objets de traitement, ou flux si l'échange consiste en flux continu d'information.

La Figure 26 montre la notation graphique utilisée pour représenter une liaison explicite qui, dans ce cas, est une liaison flux. L'objet de liaison commande et gère les interactions qui prennent place entre les interfaces d'objets de traitement qu'il embrasse. Les opérations de commande sont exercées au travers de l'interface de commande de liaison.

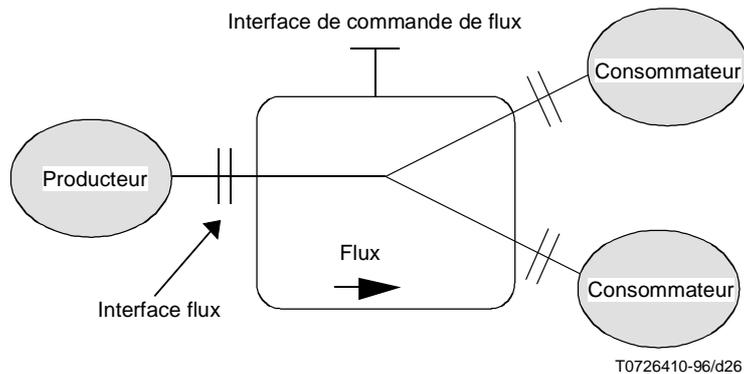


Figure 26 – Représentation d'une liaison explicite (exemple d'un flux)

La Figure 27 montre la représentation de traitement des objets de traitement qui prennent part à un échange audiovisuel dans le MMCS . On y trouve deux objets principaux: l'objet Utilisateur et l'objet Téléconférence.

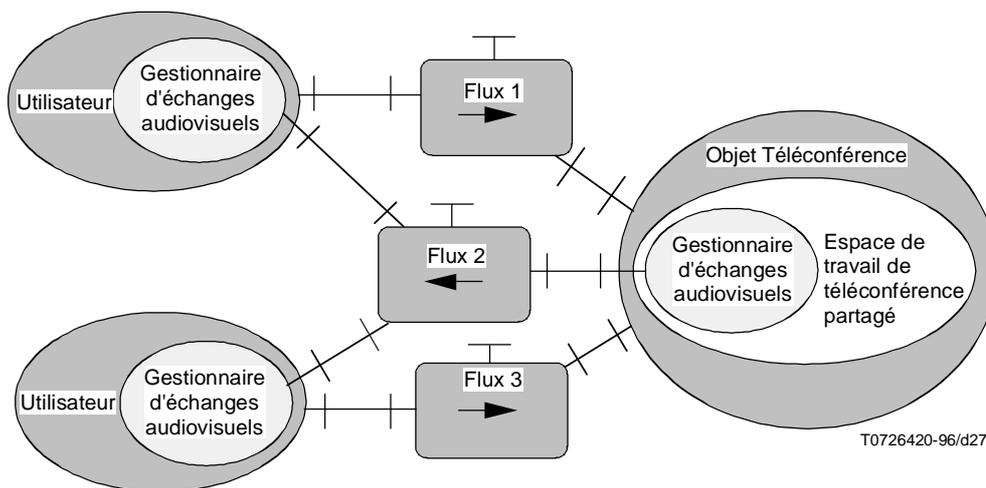


Figure 27 – Configuration MMCS d'objets impliqués dans l'échange audiovisuel

L'objet Utilisateur permet de connecter des utilisateurs finals à la téléconférence et fournit les outils qui permettent les interactions entre utilisateurs finals par l'intermédiaire de l'objet Téléconférence et des autres objets Utilisateurs. Il dote l'utilisateur final des opérations nécessaires pour se joindre à la téléconférence, pour effectuer des éditions, ou pour échanger des informations sonores et visuelles.

L'objet Téléconférence contient les fonctions requises pour réaliser une téléconférence multimédia. Il contient en particulier un espace de travail de téléconférence partagé qui porte la responsabilité de l'expédition des flux audiovisuels vers les objets utilisateurs.

Les Gestionnaires d'échanges audiovisuels sont chargés d'émettre et de recevoir les informations audiovisuelles durant la téléconférence.

Les objets Flux 1 et Flux 3 représentent les flux audiovisuels circulant des objets utilisateurs à l'espace de travail partagé. Flux 2 représente la diffusion des flux audiovisuels vers tous les objets utilisateurs.

Les interfaces de commande de flux voient passer différentes actions de commande, comme celles de gestion dynamique de la qualité de service et de synchronisation entre flux sonore et flux visuel.

### 12.1.5 Spécification d'ingénierie

Le langage d'ingénierie ODP permet de modéliser la machine de service qui prend en charge l'exécution de la spécification de traitement. La Figure 28 représente une architecture d'ingénierie simplifiée. L'élément principal de cette architecture est le noyau, qui régit l'utilisation des ressources et qui donne aux différents objets d'ingénierie les moyens de communiquer. Quelques fonctions communes à une vaste gamme de services répartis, comme les fonctions de courtage ou de gestion, sont mises à la disposition des applications réparties. Ce sont les fonctions ODP.

Le service réparti se compose d'objets d'ingénierie de base, qui sont des représentations d'exécution, sous forme, par exemple, d'un fragment de code exécutable C++, d'une spécification de traitement. Une liaison entre objets situés dans des noyaux différents se reflète dans un canal établi entre ces objets. En focalisant sur les gestionnaires audiovisuels et sur l'objet Flux 2 de la Figure 27, les moyens correspondants de prise en charge par l'ingénierie sont représentés par la Figure 28.

Un canal multipoint (représentation d'ingénierie d'un flux) est établi entre les deux gestionnaires d'échanges audiovisuels. Ce canal est relié aux noyaux concernés.

Les objets talons fournissent des fonctions d'adaptation pour la prise en charge de la transparence à la répartition (par exemple conversion de format de données entre les codages des signaux de vidéo).

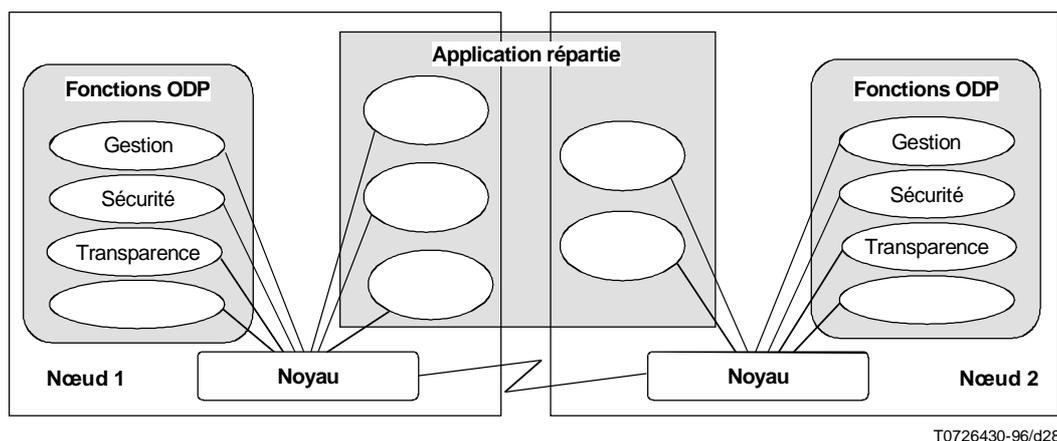
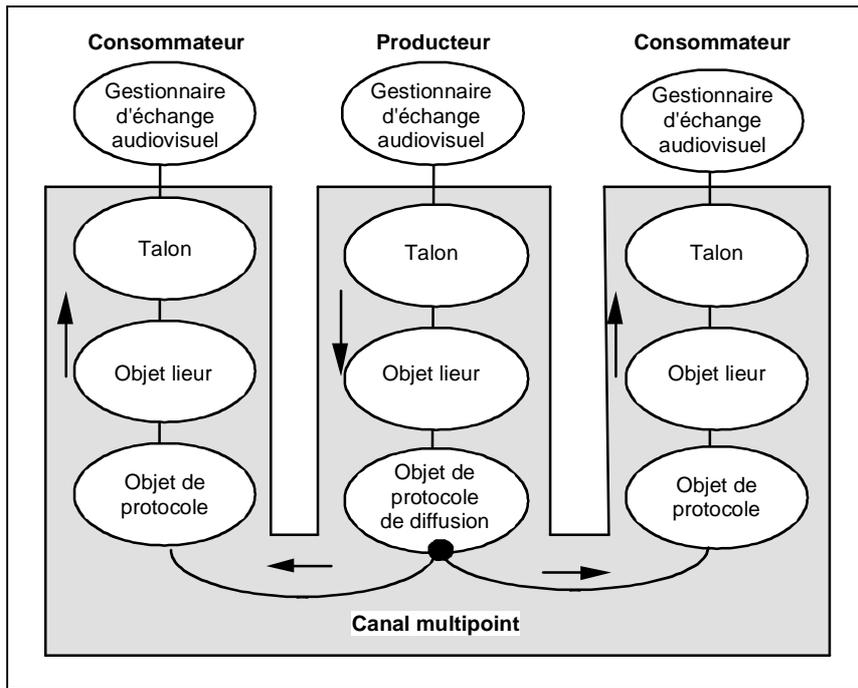


Figure 28 – Prise en charge par l'ingénierie

Les objets lieurs vérifient la compatibilité des interfaces à lier et maintiennent l'intégrité de la liaison entre les gestionnaires d'échanges audiovisuels.

Les objets de protocole coopèrent pour transporter l'information nécessaire à une prise en charge transparente à la répartition des interactions entre les gestionnaires d'échanges audiovisuels. Les objets de protocole sont choisis de manière à respecter les contraintes de qualité de service (comme, par exemple, temps réel, sécurité, etc.).



T0726440-96/d29

Figure 29 – Canal multipoint pour l'objet Flux 2

### 12.1.6 Spécification de technologie

La spécification de technologie du MMCS spécifie la réalisation du système en termes de composants matériels et logiciels. Les points de référence déterminés dans d'autres spécifications de point de vue deviennent des points de conformité où peut être vérifié le comportement du MMCS.

La spécification de technologie exprime aussi les exigences portant sur les stations de travail pour qu'elles conviennent aux représentations sonores, visuelles et textuelles. La prise en charge des besoins de communication exige l'emploi de réseaux distants rapides capables de répondre aux besoins de bande passante que demande le transport des signaux télévisuels et textuels entre des participants dispersés de par le monde.

## 12.2 Liaison flux audiovisuelle multiparties

Cet exemple traite de la spécification de la liaison flux multiparties qu'utilise le système présenté dans l'exemple précédent.

L'approche suivie dans cet exemple consiste à présenter d'abord la problématique en termes de traitement, puis de fournir les cinq spécifications de points de vue correspondantes.

On définit d'abord des concepts et des règles supplémentaires qui s'appliquent à ce problème. Dans la spécification d'entreprise, on introduit les rôles de *parties prenantes*<sup>1)</sup> (utilisateur, client, prestataire) pour mieux détailler la structure du domaine dans lequel se situe le problème. Les notations OMT [Rumbaugh 91] et IDL ont été utilisées particulièrement pour l'expression des spécifications d'information et de traitement.

Dans la spécification d'information, un certain nombre de relations sont introduites entre les classes du schéma d'invariant. Ces relations, dérivées d'un ensemble de relations de base d'OMT, sont usuellement paramétrées par du texte afin d'en préciser la signification.

<sup>1)</sup> Une partie prenante est un concept du monde des télécommunications utilisé pour représenter une organisation ou une personne qui porte un intérêt commercial ou réglementaire aux services de télécommunication.

Dans la spécification de traitement, on introduit le régulateur audiovisuel, chargé de manipuler la liaison audiovisuelle multiparties.

Dans la spécification d'ingénierie, on introduit un canal flux spécialisé pour transporter des flux continus. On montre la configuration d'objets d'ingénierie qui prend en charge la liaison audiovisuelle multiparties. Elle comporte des objets chargés de régir et de coordonner plusieurs canaux flux. Il aurait été possible de spécifier le régulateur sous une forme répartie, mais cet exemple ne comporte pas cet élément de complication supplémentaire.

La mise en œuvre que décrit le point de vue technologie sert à valider le processus de modélisation tout au long des points de vue ODP. Il est probable qu'afin d'honorer, par exemple, les critères de performances très sévères qu'exigent les applications multimédias, une solution effective ferait appel à des moyens matériels et logiciels plus spécifiques.

### 12.2.1 Description générale

L'échange d'informations continues entre applications multimédias réparties pose des problèmes compliqués. Dans l'exemple de l'application de téléconférence multimédia en temps réel, les participants, géographiquement éloignés, communiquent en échangeant des signaux sonores et visuels en temps réel. Il serait bon que l'échange audiovisuel paraisse aussi naturel et fluide que possible, ce qui demande qu'au titre de la spécification de l'application soient prises en compte des exigences relatives, par exemple, à la synchronisation des mouvements des lèvres avec le son ou à la synchronisation de l'affichage entre les diverses stations de travail.

Pour honorer ces exigences, l'application de téléconférence exprime des besoins rigoureux quant aux performances du réseau et à la synchronisation des échanges de flux audio et vidéo. Qui plus est, le nombre de flux échangés et le niveau de qualité associé à ces flux peuvent varier au cours de la téléconférence, puisque l'application dispose d'opérations de commande pour adjoindre de nouveaux participants, en retirer ou modifier le niveau de qualité de service des flux.

Pour traiter de ces fonctions complexes, le modèle de référence ODP définit dans le langage de traitement la notion d'objet de liaison. Le modèle de référence ODP donne de ce concept une description théorique dépourvue des affinements spécifiques à un type de problème particulier. Dans cet exemple, la spécification de l'objet de liaison se compose des cinq spécifications de points de vue ODP d'un objet particulier, l'objet de liaison audiovisuelle multiparties [Gay 95]. Cet objet gère les interfaces flux qui servent aux interactions audiovisuelles multiparties. Il est possible aussi d'exercer sur cet objet de liaison multiparties des opérations de commande.

La Figure 30 montre la représentation de traitement de l'objet de liaison audiovisuelle multiparties et de son environnement.

Le rectangle du centre symbolise l'objet de liaison audiovisuelle multiparties. Son environnement, représenté en grisé, est constitué de parties de l'application et du système ainsi que de l'infrastructure de réseau sous-jacente. Les symboles  $\perp$  dénotent des interfaces flux au travers desquelles les producteurs et les consommateurs d'audiovisuel échangent des signaux sonores ou (inclusivement) visuels (1). L'objet de liaison audiovisuelle multiparties gère les interactions entre les interfaces qu'il embrasse. Il cache les mécanismes qu'il utilise à cet effet et s'abstrait des aspects liés à la répartition. Le symbole  $\perp$  visible au-dessus du rectangle dénote l'interface de commande flux de l'objet de liaison. C'est au travers de cette interface que l'objet de liaison audiovisuelle multiparties fournit à son environnement des opérations (3, 4) de commande de son fonctionnement.

### 12.2.2 Spécification d'entreprise

La spécification d'entreprise produit une description des exigences et des objectifs que l'environnement impose au système à concevoir. Elle justifie la conception d'un système. Les concepts d'entreprise d'objets d'entreprise tenant des rôles d'*actions à mener* servent à la description de la liaison audiovisuelle multiparties.

Les rôles d'utilisateur, de client et de prestataire que peuvent jouer par rapport à un service les parties prenantes des télécommunications sont utilisés pour spécifier l'objet de liaison. Dans le cas de l'application et du système de soutien, il en résulte que l'on trouve des objets producteurs/consommateurs d'audiovisuel, correspondant aux utilisateurs, et des objets de gestion de ces utilisateurs, correspondant aux clients. En ce qui concerne l'infrastructure de réseau, il apparaît un objet de gestion, correspondant au prestataire, qui gère l'objet de liaison.

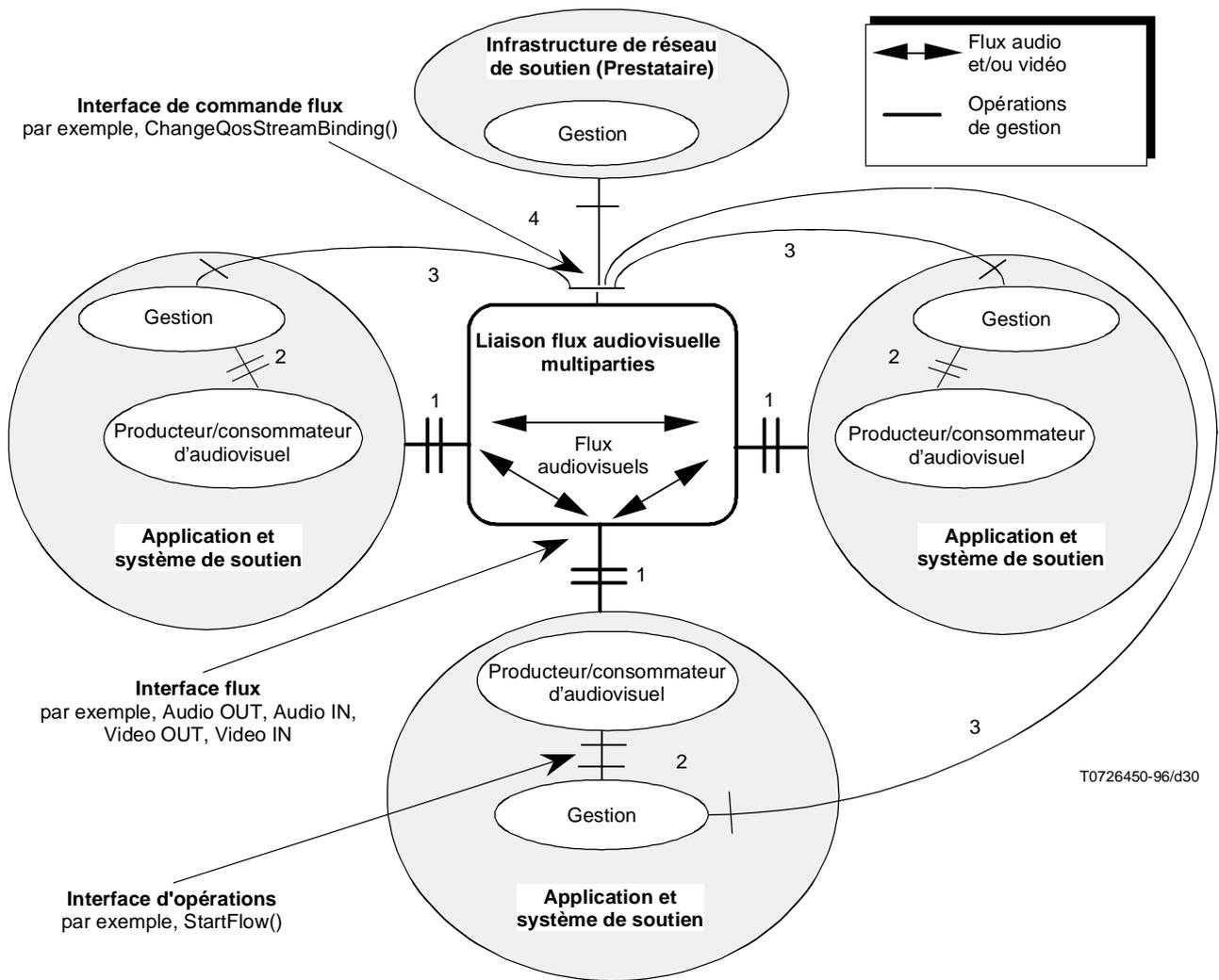


Figure 30 – Liaison flux audiovisuelle multiparties

12.2.2.1 L'utilisateur: producteur/consommateur des flux audiovisuels

Un utilisateur est un objet d'entreprise de l'application et du système de soutien qui joue un rôle dont l'objectif est de produire ou d'absorber des flux au travers de son interface flux (voir la Figure 30, 1). A titre d'exemples d'utilisateurs, on peut citer microphones, haut-parleurs, caméras, et moniteurs. Un utilisateur indique quels types de flux il est capable de traiter et quels formats de codage il exige. Il spécifie de plus les valeurs de paramètres de qualité de service qui lui sont nécessaires. Ces paramètres de qualité des flux audiovisuels peuvent prendre les valeurs télévision diffusée, télévision haute qualité, téléphonie vocale, haute fidélité ou CD. On décrit aussi des exigences de qualité de service touchant aux relations entre flux, en spécifiant, par exemple, que les mouvements des lèvres doivent être synchronisés avec les sons et qu'il faut fournir les signaux audiovisuels à plusieurs utilisateurs en simultanéité.

12.2.2.2 Le client: application locale et système de soutien

Un client est un objet d'entreprise qui joue un rôle dont l'objectif est de gérer et de prendre en charge le producteur/consommateur d'audiovisuel. Il tient compte des politiques de l'utilisateur dont il assume la gestion et la prise en charge. Les clients peuvent lancer des actions d'exécution (voir la Figure 30, 2) pour créer ou supprimer des utilisateurs ou pour ajuster les politiques d'utilisateurs existants.

Un client exécute des actions relatives à la gestion de la configuration et des ressources. Il est chargé, par exemple, de gérer les priorités entre flux: en cas de congestion dans le réseau ou de difficultés au niveau des ressources locales, le flux de plus basse priorité est arrêté ou retardé. Il exécute aussi négociations et renégociations de bout en bout avec d'autres clients pour déterminer quelles sont les valeurs acceptables, préférées ou inacceptables pour les paramètres de qualité de service. Le résultat est reflété dans le "contrat de liaison" (voir la Figure 31). Celui-ci décrit le résultat, qui est censé être honoré pendant toute l'existence de l'objet de liaison multiparties, de l'accord intervenu au terme des négociations entre utilisateur, client et prestataire de services. Le client est en mesure d'effectuer sur l'objet de liaison des actions d'exécution relatives à la gestion de la liaison flux multiparties (voir la Figure 30, 3). Ces actions traitent par exemple de la mise en route, de la suppression ou de l'ajustement d'une liaison flux en fonction des exigences de qualité négociées ou renégociées de bout en bout. L'objet de liaison est lui-même capable de lancer des actions d'exécution, par exemple afin d'indiquer qu'il n'est plus capable de respecter les valeurs négociées pour la qualité de service.

**12.2.2.3 La liaison et le prestataire de services**

La liaison est un objet d'entreprise qui rend possible l'échange des flux audiovisuels entre des utilisateurs géographiquement éloignés. Le prestataire de services est un objet d'entreprise qui offre une infrastructure de connexions qu'utilise la liaison; il gère celle-ci en fonction de l'état de l'infrastructure.

Le prestataire de services endosse la responsabilité de la topologie des flux, des questions de facturation et de sécurité, de la gestion des défaillances et de la qualité de service qu'offrent le réseau et les ressources sous-jacents. Son rôle est de gérer le réseau de telle sorte que le contrat de liaison soit honoré. Il choisit par exemple pour le canal audiovisuel l'itinéraire qui convient et lui réserve des ressources dans chacun des nœuds traversés. Si le prestataire n'est plus en mesure d'honorer le contrat de liaison, il intervient sur la liaison par des actions de négociation ou de renégociation du contrat de liaison (voir la Figure 30, 4).

**12.2.3 Spécification d'information**

La spécification d'information de la liaison audiovisuelle multiparties décrit les informations qui touchent aux parties prenantes de la liaison. Elle tient compte des exigences et des objectifs formulés dans la spécification d'entreprise.

**12.2.3.1 De la spécification d'entreprise à la spécification d'information**

Les informations qui intéressent les utilisateurs, les clients et le prestataire de services sont spécifiées dans le contrat de liaison, qui est le résultat d'un accord passé entre utilisateurs, clients et prestataire et qui satisfait aux exigences et aux objectifs issus de la spécification d'entreprise. La Figure 31 présente la relation entre les spécifications d'entreprise et d'information en termes d'OMT.

**12.2.3.2 Schéma d'invariant du contrat de liaison**

Comme le montre la Figure 31, le contrat de liaison est considéré comme un attribut de la relation entre parties prenantes. Il est décrit sous forme d'une seule classe. A un niveau de détail plus approfondi, pourtant, la spécification d'information est plus compliquée. La Figure 32 montre le schéma d'invariant du contrat de liaison audiovisuelle multiparties.

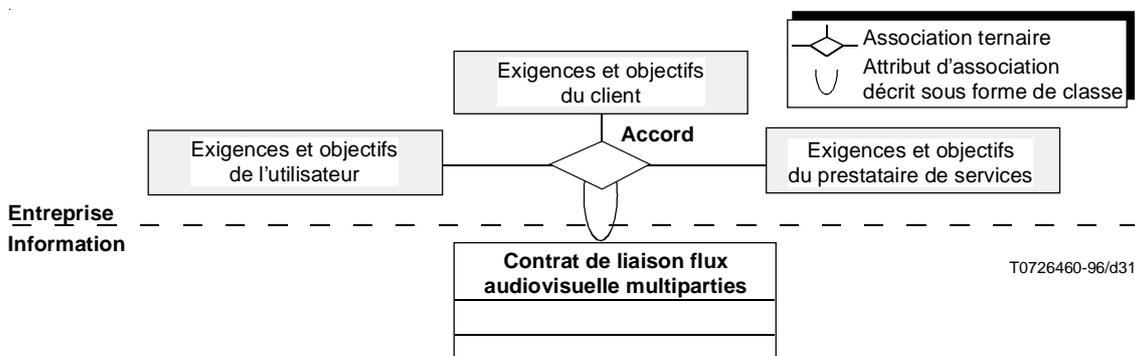
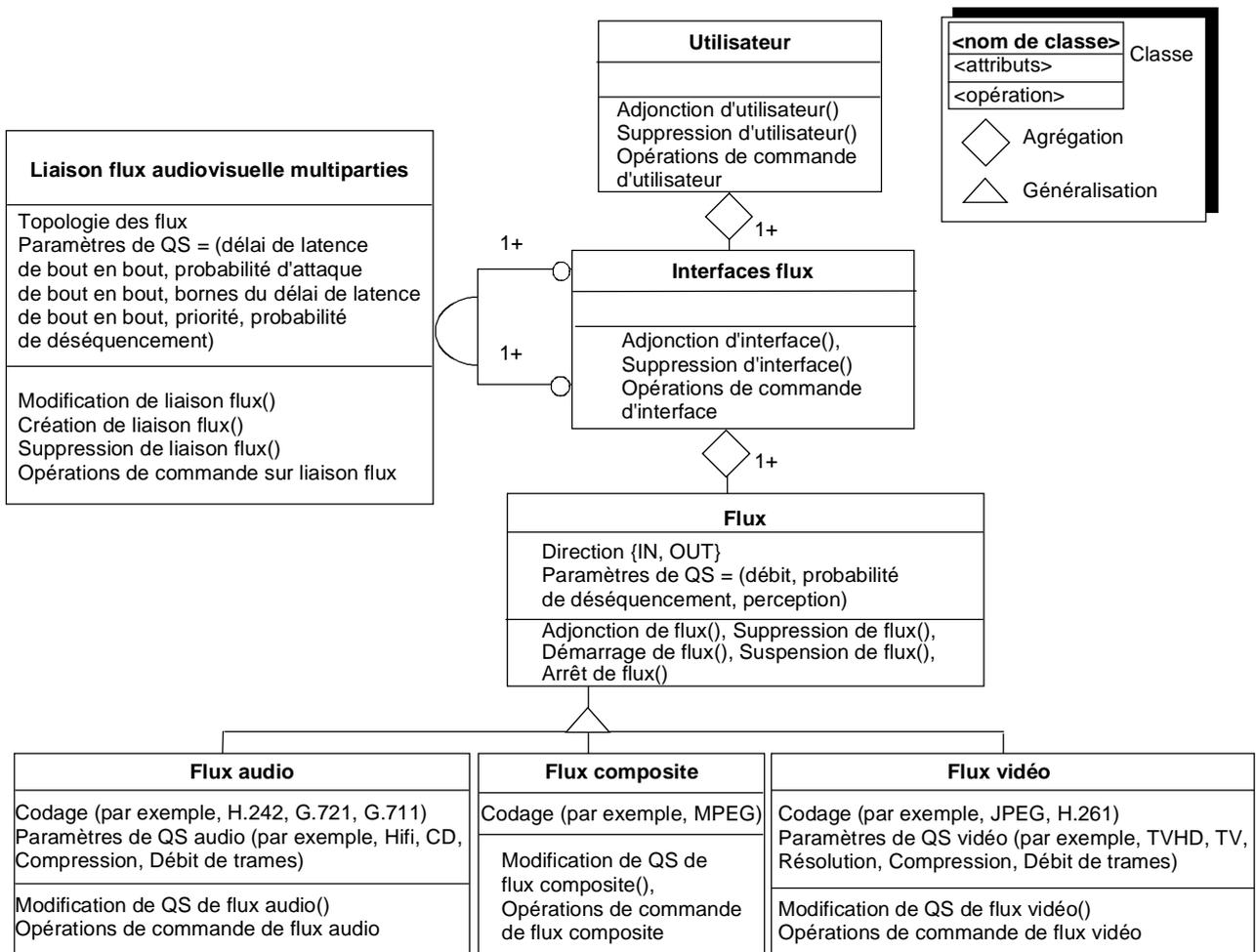


Figure 31 – Le contrat de liaison entre utilisateurs, clients et prestataire de services



T0726470-96/d32

Figure 32 – Schéma d'invariant du contrat de liaison

La structure commune à tout contrat entre clients, utilisateurs et prestataire de services est spécifiée sous la forme d'un schéma d'invariant. Dans le cas d'un contrat de liaison, on y trouve des informations sur les utilisateurs qui participent à la liaison (objets d'information Utilisateurs), sur les interfaces flux impliquées dans la liaison (objets d'information Interfaces flux) et sur les opérations que peuvent invoquer les parties prenantes. Le contrat spécifie de plus la liaison audiovisuelle multiparties qui intervient entre les interfaces flux. Les informations relatives à la liaison sont modélisées au moyen de l'objet d'information Liaison flux audiovisuelle multiparties. Le schéma d'invariant relatif à cet objet d'information spécifie toutes les opérations que les parties prenantes sont capables d'invoquer.

Il est permis à un utilisateur d'avoir une ou plusieurs interfaces flux, ce qui implique que chaque objet d'information Utilisateur se compose d'un ou de plusieurs objets d'information Interface flux. Une interface flux comporte un ou plusieurs flux, donc l'objet d'information Interface flux consiste en un ou plusieurs objets d'information Flux, audio, vidéo ou composites. Un objet d'information Flux se compose d'attributs qui indiquent, entre autres, la direction des flux et les paramètres de qualité de service. Ces derniers seront manipulés dans la spécification de traitement.

Les informations relatives à la liaison sont captées par l'objet d'information Liaison flux audiovisuelle multimédia. Cet objet relie deux ou plusieurs objets d'information Interface flux. Il contient des informations sur la topologie des flux et décrit le niveau de qualité de service qu'il est nécessaire d'entretenir durant l'échange des flux sonores et visuels entre les interfaces.

### 12.2.3.3 Schéma statique du contrat de liaison

Un contrat de liaison particulier qui existe à un moment donné entre les parties prenantes est spécifié sous forme de schéma statique. Un contrat de liaison est censé satisfaire son schéma d'invariant. Il se compose d'objets d'information Utilisateur, d'objets d'information Interface flux, d'objets d'information Flux et d'objets d'information Liaison flux audiovisuelle multiparties. D'autres objets d'information peuvent aussi participer au contrat à condition de ne pas entrer en conflit avec le schéma d'invariant.

### 12.2.3.4 Schéma dynamique du contrat de liaison

Un schéma de liaison concernant un contrat de liaison présente deux aspects: d'abord, l'effet qu'ont sur le contrat de liaison les appels d'opérations puis les conditions sous lesquelles les parties intéressées peuvent appeler ces opérations. Ces deux aspects sont étudiés ci-après.

L'effet des opérations sur le contrat de liaison dépend étroitement des choix de réalisation. On peut en général les classer en trois catégories: effet de notification, effet de négociation et sans effet.

Une opération qui porte effet de notification est une opération par laquelle une partie prenante informe le contrat de liaison de la création d'un nouvel objet de traitement. L'effet sur le contrat est l'adjonction d'un nouvel objet d'information. L'opération Adjonction de nouvel utilisateur, par exemple, a pour résultat la création d'un nouvel objet d'information Utilisateur, d'un ou de plusieurs objets d'information Interface flux et d'un ou de plusieurs objets d'information Flux.

Une opération qui porte effet de négociation est une opération par laquelle une partie prenante négocie avec l'objet de liaison et avec les autres parties prenantes une modification du contrat de liaison. Le contrat est modifié si la négociation réussit. Une opération réussie de Suppression d'utilisateur, par exemple, entraîne l'élimination d'un objet d'information Utilisateur et des objets informationnels associés de flux et d'Interface flux. De plus, l'objet d'information de liaison qui relie les interfaces de l'utilisateur supprimé aux interfaces d'autres utilisateurs sera modifié ou enlevé.

Une opération qui ne porte aucun effet est une opération qui n'affecte pas les objets d'information du contrat de liaison. Il s'agit essentiellement d'opérations de commande. L'opération Suspension de flux, par exemple, aura pour effet de suspendre le flux concerné, mais sera sans conséquences sur l'information relative au contrat de liaison.

Le schéma dynamique décrit aussi les conditions sous lesquelles les opérations qui portent sur le contrat de liaison peuvent être invoquées. Les schémas d'invariant et statique n'imposent pas d'ordre particulier entre les opérations que peuvent appeler les parties prenantes. Pour maintenir cependant un contrat de liaison sensé, il est nécessaire de définir des conditions concernant l'appel des opérations. En voici quelques exemples:

- un client ne peut invoquer d'opération d'adjonction de nouvel utilisateur que sur une liaison flux existante;
- les clients ne peuvent invoquer les opérations de changement de QS de flux sonore ou de suppression de flux que sur des flux existants;
- les clients et les prestataires de services ne peuvent invoquer d'opération de modification de liaison flux que sur une liaison existante.

## 12.2.4 Spécification de traitement

Le présent paragraphe donne la signature de l'interface de commande de la liaison flux. Pour avoir une description complète de cette interface, il faudrait aussi décrire les contrats de comportement et d'environnement.

### 12.2.4.1 De la spécification d'information à la spécification de traitement

Le mappage des objets d'information sur les objets de traitement n'est pas nécessairement un pour un. Il faut le spécifier dans chaque cas de telle sorte que soit assurée la cohérence entre les spécifications. Le regroupement des classes en objets est une décision qui appartient au concepteur du système. Il n'est pas nécessaire à cette étape de tenir compte des effets de la répartition.

La Figure 33 illustre la projection de plusieurs classes d'information sur des interfaces de traitement. Pour l'interface de commande de la liaison audiovisuelle multiparties (voir la Figure 33, 3 et 4), les opérations de la classe de liaison flux audiovisuelle multiparties sont prises en compte. L'interface flux d'opérations du producteur/consommateur d'audiovisuel, 2, sera reflétée dans les opérations décrites dans la classe Flux. L'interface flux, 1, possède les caractéristiques des attributs de la sous-classe Flux. Les attributs exprimés dans la spécification d'information seront représentés par des paramètres des opérations de traitement. Les appellations des opérations et des paramètres des deux points de vue sont indépendantes.

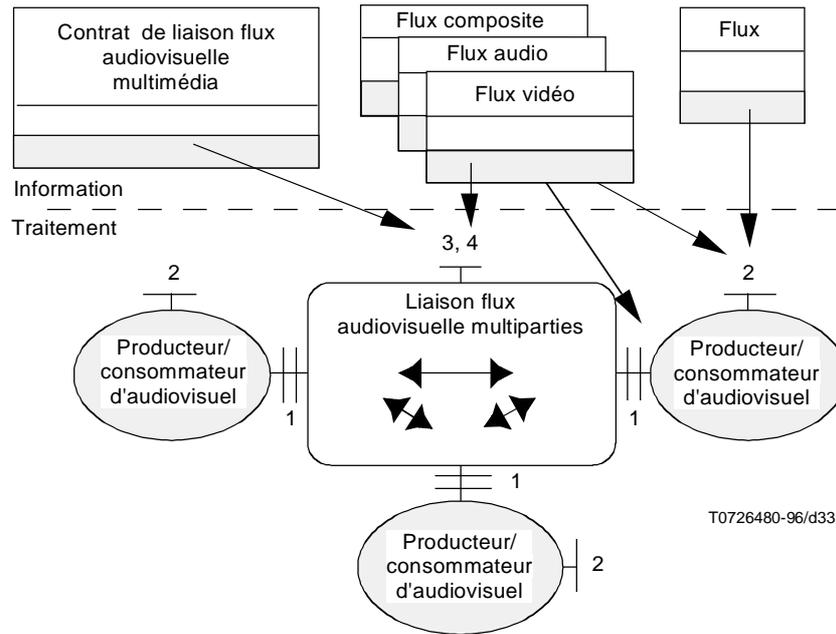


Figure 33 – Projection de l'information sur le traitement

12.2.4.2 Spécification en IDL

NOTE – Ce qui suit est très proche de l'IDL de l'OMG [OMG IDL].

Le modèle de référence ODP décrit un modèle de traitement applicable aux applications réparties, mais il ne fournit pas de langage de spécification particulier pour les objets et les interfaces de traitement. C'est pourquoi il est fait appel ici à un langage de spécification supplémentaire, l'IDL, pour déduire de l'interface de commande de la liaison flux une spécification de traitement (voir le Tableau 1). L'IDL donne les moyens d'exprimer des spécifications de traitement ayant une connotation télécommunications et multimédia. Les spécifications IDL sont déduites de la spécification d'information.

Tableau 1 – Spécification IDL de l'interface de commande de liaison flux

```

interface template StreamBindingControlInterface; /* (3), (4) type d'interface d'opération */
typedef sequence <Flow> StreamInterface;
operations
void ChangeQosStreamBinding (in StreamBindingId Binding,
    in QOS RequestedQos, out QOS ProvidedQos);
void RemoveStreamBinding (in StreamBindingId Binding,
    out StreamBindingId RemainingBindings);
void AddNewUser (in AVuserId Newuser, in StreamInterface NewFlows,
    in QOS RequestedQos, out QOS ProvidedQos, out ResultReport StatusBinding);
/* D'autres opérations de commande de liaison flux que détermineraient l'application, le système ou la partie gestion de
réseau, sont possibles. */
/* behaviour 'Une occurrence de ce gabarit d'interface fournit d'autres objets de traitement pour exécuter des actions de
commande sur l'objet de liaison multiparties.' */
    
```

### 12.2.4.3 Choix de traitement pour la configuration d'échange audiovisuel multiparties

Il est possible d'affiner la liaison flux audiovisuelle multiparties pour arriver à une configuration d'ingénierie. Plusieurs solutions sont possibles, mais celle qui a été choisie sert dans de nombreux systèmes multimédias répartis. Il existe dans plusieurs réalisations de systèmes répartis multiparties un composant fonctionnel, appelé régulateur audiovisuel, qui gère les flux audiovisuels. Il reçoit les flux audiovisuels des producteurs et les réfléchit, après manipulation éventuelle, vers les consommateurs. C'est cette approche qui a été adaptée dans la Figure 34 à la liaison flux audiovisuelle multiparties.

Le régulateur audiovisuel est chargé de la redirection des opérations de commande de flux 3 et 4 vers chacune des liaisons de sous-flux 5. Il s'occupe aussi de l'établissement, de la commande et de la libération des liaisons audiovisuelles entre producteurs et consommateurs. Il négocie les exigences et les objectifs, relatifs par exemple aux codages ou aux débits de trames, déterminés par la spécification d'entreprise. Une liaison flux audiovisuelle relie chacun des objets producteur/consommateur d'audiovisuel au régulateur audiovisuel.

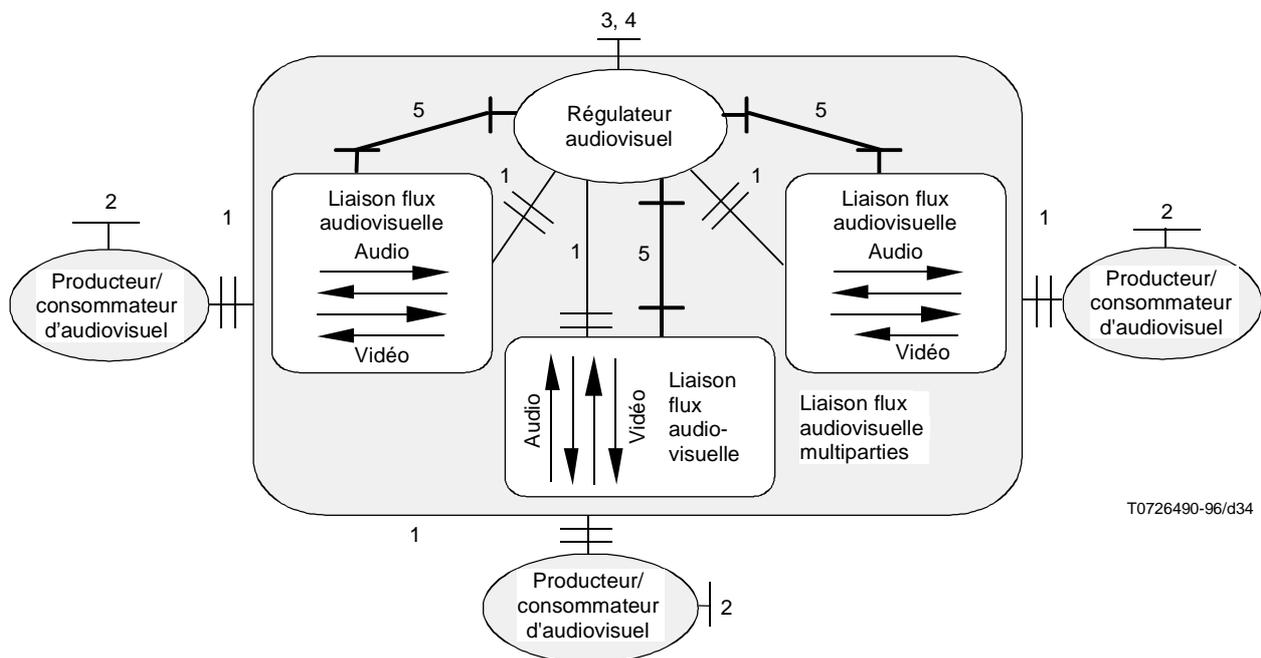


Figure 34 – Objets impliqués dans un échange audiovisuel multiparties

### 12.2.5 Spécification d'ingénierie

Le présent paragraphe donne une spécification d'ingénierie pour l'objet liaison audiovisuelle multiparties que définit la spécification de traitement.

#### 12.2.5.1 De la spécification de traitement à la spécification d'ingénierie

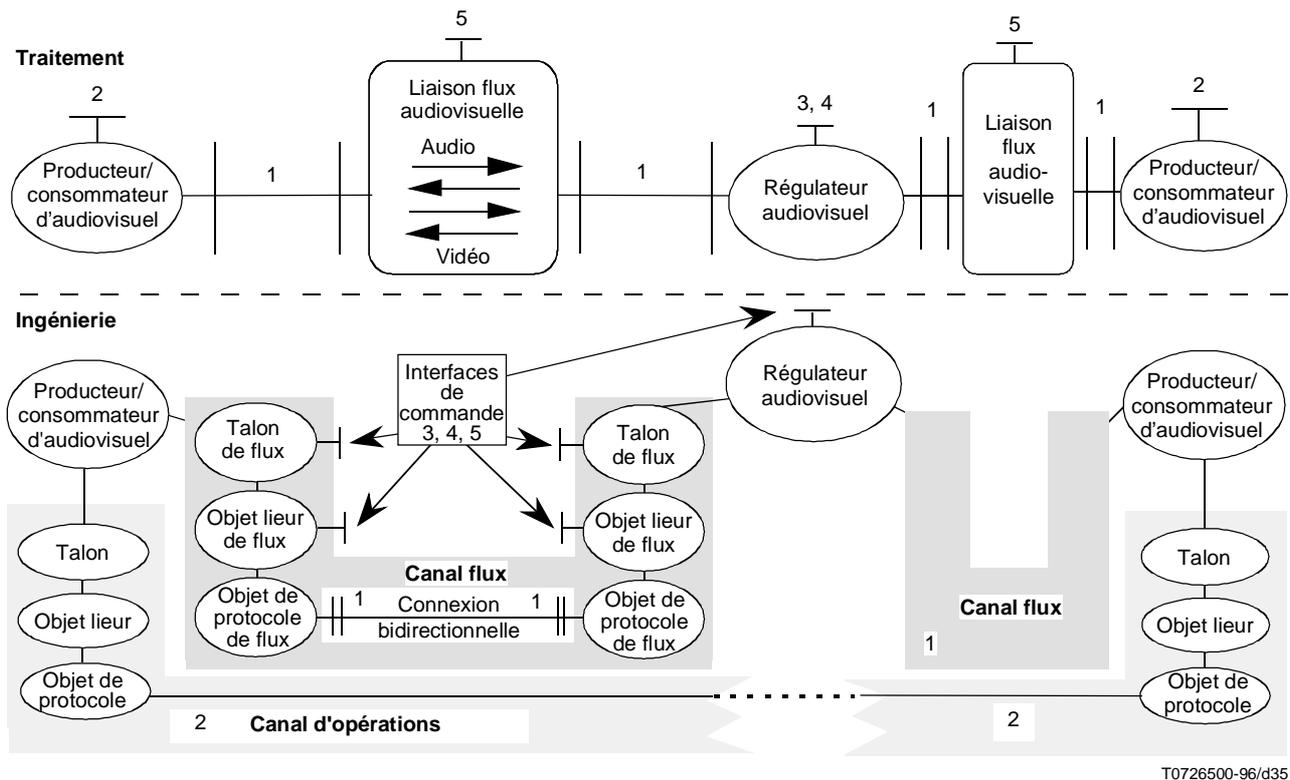
Il faut projeter la spécification de traitement sur une spécification d'ingénierie. Cette spécification d'ingénierie préserve le comportement décrit dans la spécification de traitement.

La transformation en ingénierie d'une interface flux (voir la Figure 35, 1) conduit à la création d'un canal flux producteur/consommateur spécialisé en flux continu. Les paramètres de qualité de service associés au flux audiovisuel défini dans la spécification de traitement influent sur le choix des composants du canal.

Une interface d'opération (voir la Figure 35, 2) se reflète dans la spécification d'ingénierie sous forme d'une configuration de canal client-serveur en accord avec la définition donnée par le modèle de référence ODP. Lors de l'établissement d'un canal entre les objets de traitement concernés, les contrats d'environnement spécifiques aux interfaces, pour la sécurité, par exemple, sont pris en compte.

Des interfaces de traitement de commande de liaison flux (voir la Figure 35, 3, 4 et 5) apparaissent dans la représentation d'ingénierie dans différents nœuds. Leurs communications s'effectuent au travers de canaux de commande, non figurés dans la Figure 35, dont la structure est la même que celle des canaux d'opérations. Il est permis de plus de créer des objets de soutien, comme par exemple des objets de synchronisation, chargés de gérer et de régenter un ensemble de canaux exerçant des relations mutuelles.

L'objet producteur/consommateur d'audiovisuel ainsi que l'objet régulateur d'audiovisuel sont transformés en objets d'ingénierie de base. Si ces objets se trouvent répartis au travers de nœuds différents, il faut décomposer plus avant et créer plusieurs objets d'ingénierie et parties de canaux. La Figure 35 illustre le mappage de deux producteurs/consommateurs d'audiovisuel de traitement sur une spécification d'ingénierie.



T0726500-96/d35

Figure 35 – Mappage du traitement sur l'ingénierie

### 12.2.5.2 Le canal flux d'ingénierie

Le concept de canal de l'ODP fournit les mécanismes d'ingénierie qui assurent la transparence à la répartition des interactions entre objets d'ingénierie de base. Le canal se compose de trois objets d'ingénierie, à savoir l'objet de protocole, l'objet lieur et le talon. Les canaux transportent deux types d'information différents: les opérations de commande, tout d'abord, qui permettent par exemple de négocier la qualité de service, n'ont besoin que de peu de bande passante, mais exigent une fiabilité élevée; ensuite, les interactions en temps réel, du genre échange de signaux vocaux et vidéo, qui consomment plus de bande passante mais demandent une fiabilité moindre. Les canaux sont donc divisés en canaux d'opérations et canaux flux, dont les caractéristiques diffèrent. Le présent paragraphe présente d'abord les concepts d'objets talons, lieurs et de protocole avant d'expliquer comment s'établit un canal flux particulier.

L'**objet talon** apporte des fonctions d'adaptation destinées à prendre en charge les interactions entre interfaces d'objets d'ingénierie de base résidant dans des nœuds différents. Pour les appels d'opérations, il se charge de rassembler ou de disperser les paramètres des opérations pour donner aux interactions la transparence d'accès. Du fait de la différence de nature des informations échangées, les flux ont besoin de trouver d'autres fonctions dans le talon: celui-ci doit pouvoir offrir des mécanismes de codage et de décodage des informations sonores et visuelles. Le talon doit de plus être capable de notifier au producteur/consommateur d'audiovisuel que des données sont disponibles. Les objets talons de flux fournissent aussi des opérations pour manipuler les ressources locales (augmentation, par exemple, de la taille des mémoires tampons) et pour notifier des événements relatifs au flux comme, par exemple, modifications de qualité de service, manque d'espace dans les mémoires tampons, perte de données, etc. Un objet talon dispose d'une interface de présentation qu'emploie l'objet qui est lié au canal et une interface de commande pour, par exemple, la gestion de la qualité de service.

L'interaction réciproque entre **objets lieurs** a pour but de préserver l'intégrité de la liaison. Ces objets enregistrent des informations relatives au canal. Ils portent aussi la responsabilité de la validation des références d'interfaces et de l'interaction avec l'objet de relocalisation pour recouvrer les informations sur la position de l'interface à la suite d'une erreur de liaison. Dans le cas des flux, les objets lieurs conservent des informations concernant la qualité de service requise. Un objet lieur dispose d'une interface de commande qui permet de modifier la configuration du canal et de le détruire pour partie ou en totalité.

Les **objets de protocole** assurent l'interaction à distance entre les objets de traitement. Leur présence est nécessaire lorsque les objets de traitement qu'il faut lier sont situés dans des nœuds différents. On utilise en général le mécanisme RPC, qui envoie à un objet de protocole distant capable de l'accueillir, une opération mise sous forme de message. L'objet auquel est destiné l'appel exécute la procédure et retourne un message de réponse au demandeur. Dans le cas de la liaison flux audiovisuelle multiparties, les objets de protocole de type RPC conviennent aux objets de traitement qui appellent des opérations, comme dans le cas des canaux de commande et de gestion (voir la Figure 35, 3, 4 et 5). Mais dans le cas d'échange de flux continus, il faut faire appel à un autre protocole spécialisé, le protocole de flux, qui n'emploie pas le mécanisme du RPC. Celui-ci demande que chaque donnée de mémoire tampon à transférer soit associée à une action distincte (en d'autres termes, il n'existe aucune relation particulière entre appels RPC passés et à venir). Comme les flux continus impliquent des relations entre appels, on établit un protocole de flux qui crée un canal virtuel entre deux objets de protocole de flux pour la durée de l'échange du flux audiovisuel. Il est possible dans ce cas de définir des relations spécifiques entre données sonores et visuelles. Les objets de protocole peuvent entrer en interaction avec des objets situés hors du canal, un courtier par exemple, afin d'en obtenir les informations dont ils ont besoin.

### 12.2.6 Spécification de technologie

Le présent paragraphe décrit le début d'une mise en œuvre d'une liaison flux utilisant ANSAware, qui est fondée sur une architecture semblable à celle d'ODP. On emploie ANSAware car c'est une plate-forme répartie présentant l'avantage de réaliser plusieurs mécanismes d'ingénierie (ANSAware, par exemple, prend en charge les interfaces d'opération). La spécification de technologie présentée ici correspond à la mise en œuvre d'un service de visiophonie multiparties grâce auquel des utilisateurs finals sont en mesure d'échanger des informations audiovisuelles à partir de leurs ordinateurs de bureau. Il est possible de faire entrer de nouveaux utilisateurs ou d'en retirer pendant le cours de la session.

La spécification de technologie découle des spécifications sous les autres points de vue, mais la réalisation est principalement fondée sur la spécification d'ingénierie. La spécification de technologie consiste en une description des matériels et des logiciels qui sont capables de mettre en œuvre la spécification d'ingénierie tout en tenant compte d'exigences d'entreprise supplémentaires comme, par exemple, la mise à disposition de matériels et de logiciels.

La Figure 36 montre un mappage d'objets d'ingénierie sur des solutions techniques. Elle est focalisée sur la mise en œuvre du canal d'opérations et du canal flux et montre comment réaliser les objets d'ingénierie sous forme d'éléments matériels et logiciels.

## 12.3 Exemple de gestion – Objet métrique

L'ISO/CEI 10164-11 (objets et attributs métriques) donne la spécification d'un objet métrique de surveillance de moyenne qui examine dans un objet déterminé une certaine valeur d'attribut à des intervalles de temps réguliers dépendant d'une période d'échantillonnage. L'objet met à jour une estimation de la moyenne mobile de la valeur de l'attribut observé et applique sur cette moyenne un mécanisme de seuil qui déclenche l'émission d'une notification en cas de dépassement de ce seuil.

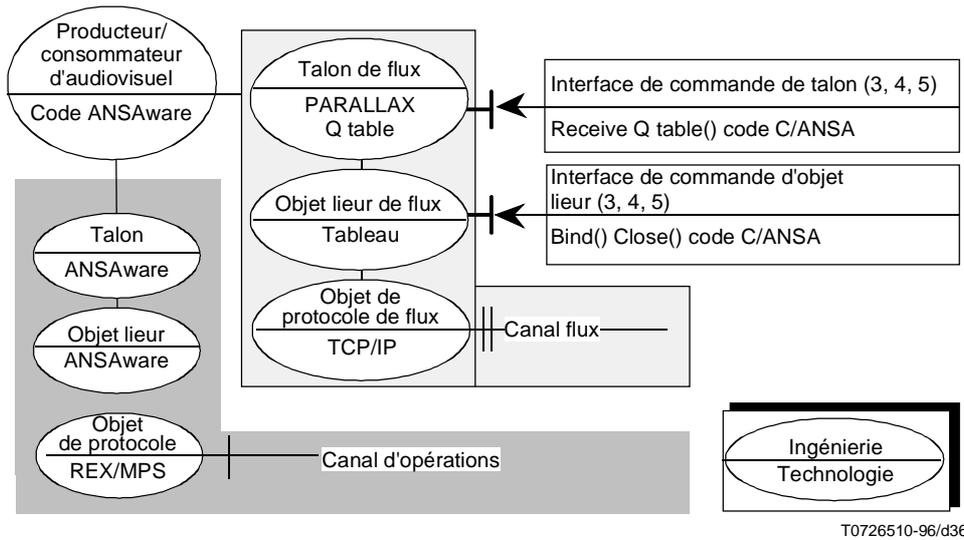


Figure 36 – Projection de l'ingénierie sur la technologie

### 12.3.1 Spécification d'entreprise

#### Communautés:

Communauté d'opération de mesure, mettant en jeu les rôles suivants:

- rôle d'examineur – rôle d'objet client qui assume la responsabilité du déclenchement des examens portant sur un objet observé associé jouant le rôle d'observé;
- rôle d'observé – rôle d'objet serveur qui répond à l'examen d'une valeur d'attribut observée.

Communauté de commande de mesure, mettant en jeu les rôles suivants:

- rôle de directeur de mesure – rôle d'objet client qui gère l'objet responsable du déclenchement des opérations de mesure pour un ou plusieurs objets métriques;
- rôle de commande de mesure – rôle d'objet serveur pour l'objet métrique qui répond aux opérations de commande destinées à modifier l'état de l'objet métrique par référence aux paramètres de l'algorithme associé au comportement de mesure.

Communauté de notification de mesure, mettant en jeu les rôles suivants:

- rôle d'annonceur de mesure – rôle d'objet client de l'objet métrique qui émet des notifications en fonction de l'algorithme de mesure agissant sur les valeurs examinées d'un attribut observé;
- rôle de distribution de notification – rôle d'objet serveur qui reçoit des notifications émises en vue de leur distribution ultérieure.

#### Politiques:

Politique d'examineur – l'objet examineur a l'obligation de déclencher des examens en respectant des intervalles de temps réguliers dont la valeur est établie par l'attribut de commande de période d'échantillonnage.

Politique d'observation – l'objet observé a l'obligation de répondre en donnant la valeur observée avant que ne s'écoule une période d'échantillonnage depuis l'arrivée de la commande d'examen. S'il n'est pas en mesure de répondre, une erreur est indiquée et l'observation non validée pour cet examen.

Politique de commande de mesure – l'objet métrique a l'obligation de refléter dans le comportement de traitement de son algorithme de moyenne mobile les modifications des paramètres de l'algorithme effectuées par le truchement du rôle de commande de mesure. Il est nécessaire de régir les paramètres suivants de l'algorithme: période d'échantillonnage, période de moyenne mobile, pointeur de relation d'instance d'objet observé, identifiant de l'attribut observé, seuil de déclenchement de notification, seuil de réarmement.

Politique de notification de mesure – l'objet métrique a l'obligation d'émettre une notification lorsque la moyenne mobile de la valeur observée traverse le seuil de déclenchement de notification. Un mécanisme d'hystérésis fournit un seuil de réarmement qui, lorsqu'il est franchi à la descente, réarme le seuil supérieur aux fins de notification ultérieure.

### 12.3.2 Spécification d'information

Les objets d'information les plus intéressants sont l'objet métrique et l'objet observé.

**description d'information d'objet métrique de surveillance de moyenne {**

**Attributs:**

**identifiant d'attribut observé** – identifiant à employer à l'obtention de la valeur de l'attribut observé.

**période d'échantillonnage** – intervalle de temps entre deux examens successifs de la valeur de l'attribut observé de l'objet observé associé.

**période de moyenne mobile** – durée effective au cours de laquelle est effectué le calcul de la moyenne mobile de la valeur de l'attribut observé.

**jauge dérivée** – valeur courante de la moyenne mobile telle qu'elle résulte de l'examen de l'objet métrique et du comportement de mise à jour algorithmique.

**seuil de déclenchement de notification** – valeur à comparer à la jauge dérivée à la fin de chaque mise à jour algorithmique avec émission d'une notification lorsque la détente est armée et que la valeur de la jauge dérivée dépasse la valeur de déclenchement. Le déclencheur est désarmé par la notification, jusqu'à réarmement ultérieur par application du mécanisme d'hystérésis.

**seuil de réarmement** – valeur à comparer à la jauge dérivée à la fin de chaque mise à jour algorithmique avec réarmement de la détente lorsque la valeur de la jauge dérivée est inférieure à celle du seuil de réarmement.

**Etats:**

**état opérationnel** – reflète le fait que l'objet métrique fonctionne ou ne fonctionne pas.

**état examen** – ses valeurs sont:

en attente (du démarrage de l'examen en fin d'intervalle d'échantillonnage)

en examen (en attente du résultat de l'examen de l'objet observé)

**état armement de seuil** – ses valeurs sont armé, désarmé

**Invariants:**

la période de moyenne mobile doit être supérieure à la période d'échantillonnage.

la valeur du seuil de réarmement doit être inférieure à la valeur du seuil de déclenchement de notification.

**Transitions d'état:**

L'état opérationnel ayant la valeur habilité, les transitions d'état d'information suivantes peuvent se produire:

en attente -> en examen – se produit lorsque l'objet métrique atteint la fin de la période d'échantillonnage.

en examen -> en attente – se produit:

a) lorsque la valeur de l'attribut observé a été renvoyée et que l'algorithme a mis à jour la valeur dérivée et a appliqué le processus de traitement des seuils; ou

b) lorsque l'examen est invalidé.

armé -> désarmé – se produit lorsque le processus de traitement des seuils détecte la traversée du seuil de déclenchement de notification, ce qui déclenche l'émission d'une notification.

désarmé -> armé – se produit lorsque le processus de traitement des seuils détecte la traversée du seuil de réarmement.

**Relations:**

**Instance d'objet observé – pointeur vers l'objet observé dont l'identifiant d'attribut observé est examiné à chaque instant d'échantillonnage.**

}

L'objet d'information "objet observé" est tout objet disposant d'un attribut identifiable ayant une valeur examinable. Le seul état intéressant de l'objet observé est la valeur actuelle de ses attributs observés.

**description d'information d'objet observé {**

**Attributs:**

**attribut observé – tout attribut identifié dont le type de valeur est entier ou réel.**

}

L'objet d'information suivant donne un modèle pour le contenu d'un objet d'enregistrement d'alerte de qualité de service, associé à un événement de traversée positive du seuil de déclenchement de notification.

**description d'information d'enregistrement d'alerte de qualité de service {**

**Attributs:**

**identifiant d'instance d'objet métrique**

**identifiant d'instance d'objet observé**

**identifiant d'attribut observé**

**instant de notification**

**valeur de jauge dérivée**

}

### 12.3.3 Spécification de traitement

La localisation d'un objet métrique dans un système différent de celui qui abrite l'objet observé sort du champ de la norme de fonction de gestion-systèmes d'objets métriques.

Les gabarits suivants d'objets et d'interfaces de traitement montrent comment décomposer les objets de traitement pour permettre de séparer un objet métrique de l'objet observé qu'il examine.

**objet de traitement métrique de surveillance de moyenne {**

**Interface serveur:**

**commande de surveillance de moyenne – serveur**

**Interfaces clients:**

**valeur d'examen d'objet observé – client**

**alerte de qualité de service – client**

**COMPORTEMENT – voir l'ISO/CEI 10164-11**

}

**Gabarits d'interfaces:**

**interface de commande de surveillance de moyenne {**

**OPÉRATIONS (attributs, qui tous sont munis de méthodes d'acquisition et de remplacement):**

**instance d'objet observé**

**identifiant d'attribut observé**

**période d'échantillonnage**

**période de moyenne mobile**

jauge dérivée  
 seuil de déclenchement de notification  
 seuil de réarmement  
 état opérationnel

Contrat d'environnement:

les opérations de commande doivent agir sur le comportement paramétrisé de l'algorithme de mesure pour les examens lancés après que les valeurs d'attributs ont été établies.

}

interface de valeur d'examen d'objet observé {

OPÉRATIONS:

le client lance une opération d'acquisition sur l'instance d'objet observé afin d'obtenir la valeur courante de l'attribut observé.

Contrat d'environnement:

si l'opération d'examen n'a pas eu pour résultat le retour de la valeur observée avant le début du prochain cycle d'échantillonnage, alors l'examen est déclaré non valable et son résultat n'est pas utilisable dans le processus de mise à jour par l'algorithme de mesure.

}

interface d'alerte de qualité de service {

OPÉRATIONS:

"ISO/CEI 10164-4": notification d'alerte de qualité de service (*qualityOfServiceAlarmNotification*)

Contrat d'environnement:

les valeurs de la notification d'alerte de qualité de service sont établies aux valeurs présentées dans l'enregistrement d'alerte de la spécification d'objet d'information.

}

## 12.4 Exemple de base de données

Cet exemple est fondé sur un système (très simplifié) de traitement de commandes qui prend en charge les activités commerciales d'un organisme de vente, qui peut n'être qu'une partie d'une structure plus importante, s'occupant de la fourniture de certains produits à des clients.

L'organisme de vente dispose d'un certain nombre d'entrepôts où sont stockées des gammes de produits. Ces produits sont vendus à des clients, qui sont des organisations extérieures reconnues comme partenaires commerciaux dignes de confiance financière. Une vente se compose du placement d'une commande par un client pour une quantité donnée d'un ou de plusieurs produits, de la fourniture au client des produits demandés à partir d'un entrepôt, de la facturation au client du prix des produits et, finalement, de l'enregistrement du paiement par le client.

### 12.4.1 Spécification d'entreprise

Pour ce système, on peut faire apparaître les objets d'entreprise suivants:

- organisme de vente: communauté qui a le pouvoir de commercer;
- liaison client: entité appartenant à l'organisme de vente qui permet aux clients de placer leurs commandes;
- gestion d'entrepôt: entité appartenant à l'organisme de vente qui exerce la responsabilité du fonctionnement d'un entrepôt;
- comptes: entité appartenant à l'organisme de vente qui exerce la responsabilité des questions financières;
- traitement de commande: entité appartenant à l'organisme de vente dont la responsabilité est de traiter les commandes et d'en conserver la trace;
- client: organisation qui commerce avec l'organisme de vente en lui passant des commandes.

La Figure 37 donne des exemples de ces objets d'entreprise.

**12.4.2 Spécification d'information**

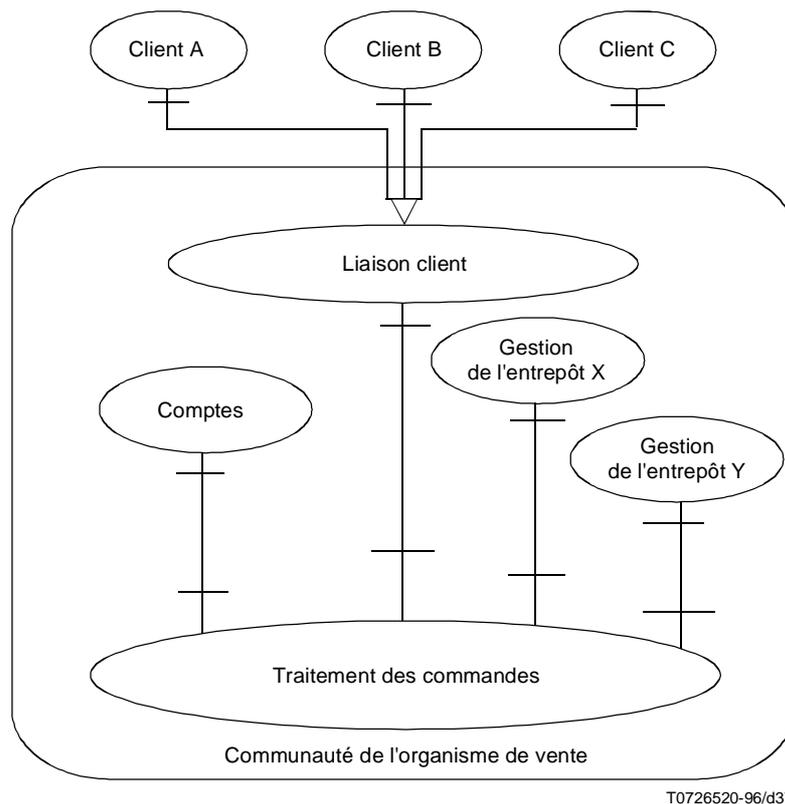
La Figure 38 propose un schéma d'invariant qui peut convenir pour ce système. Il est exprimé par usage de la notation OMT avec des exemples d'attributs et d'opérations. Il représente l'état et le comportement des objets d'information qui relèvent de l'organisme de vente ainsi que leurs relations réciproques. Il existe en particulier un objet d'information Client qui représente les informations relatives à l'objet d'entreprise Client dont a besoin l'organisme de vente pour entretenir ses relations commerciales avec les organisations clientes.

**12.4.3 Spécification de traitement**

La spécification de traitement relative à cet exemple possède deux caractéristiques fondamentales:

- il existe de nombreux utilisateurs qui correspondent chacun à un des objets d'entreprise (liaison client, gestion d'entrepôt et comptes) et qui requièrent chacun des services qui se rapportent à une certaine partie du schéma d'information;
- toutes les occurrences du schéma d'information doivent avoir une représentation partagée permanente.

Un système de base de données est capable de répondre à ces deux exigences; la Figure 39 propose un modèle de traitement qui peut convenir à cet exemple.



**Figure 37 – Système de traitement de commandes – Spécification d'entreprise**

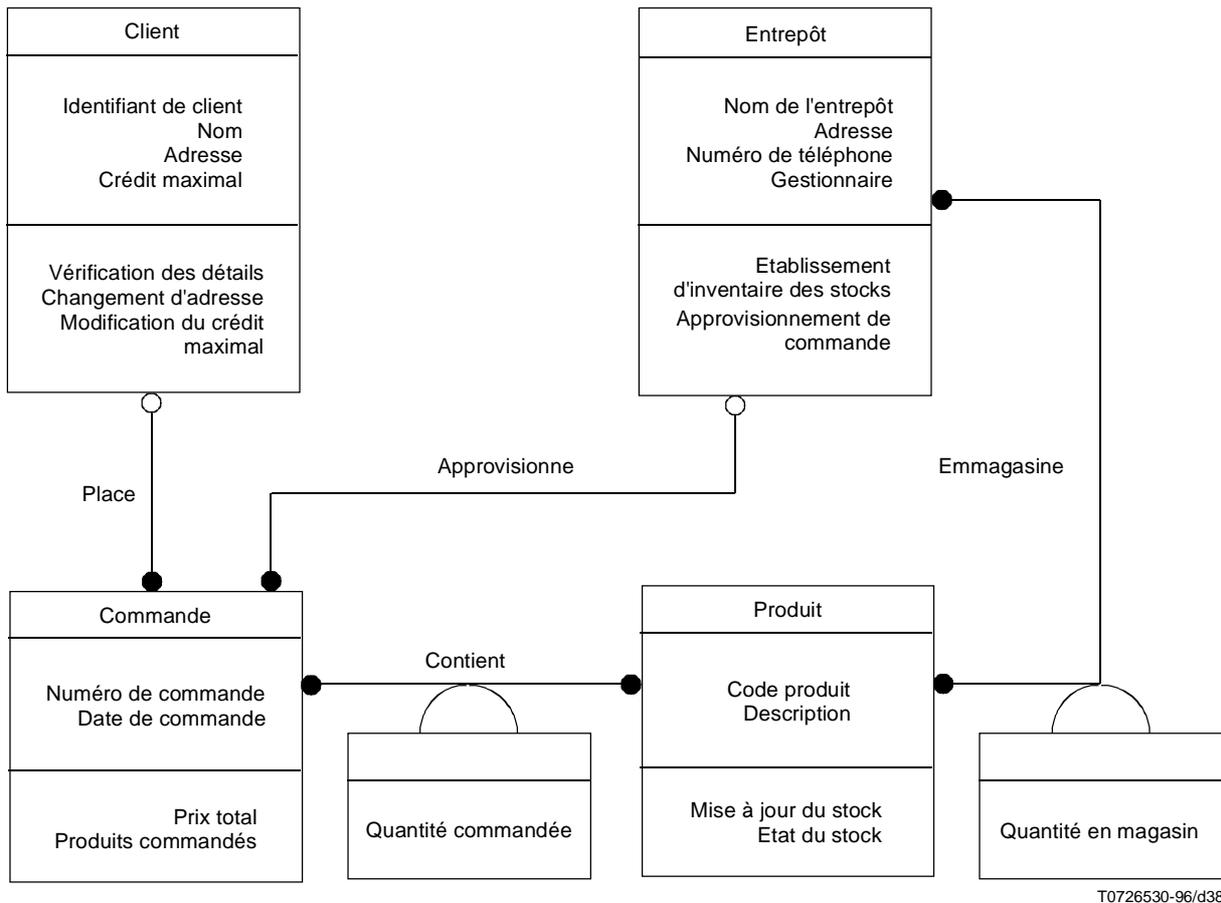


Figure 38 – Système de traitement de commandes – Spécification d'information

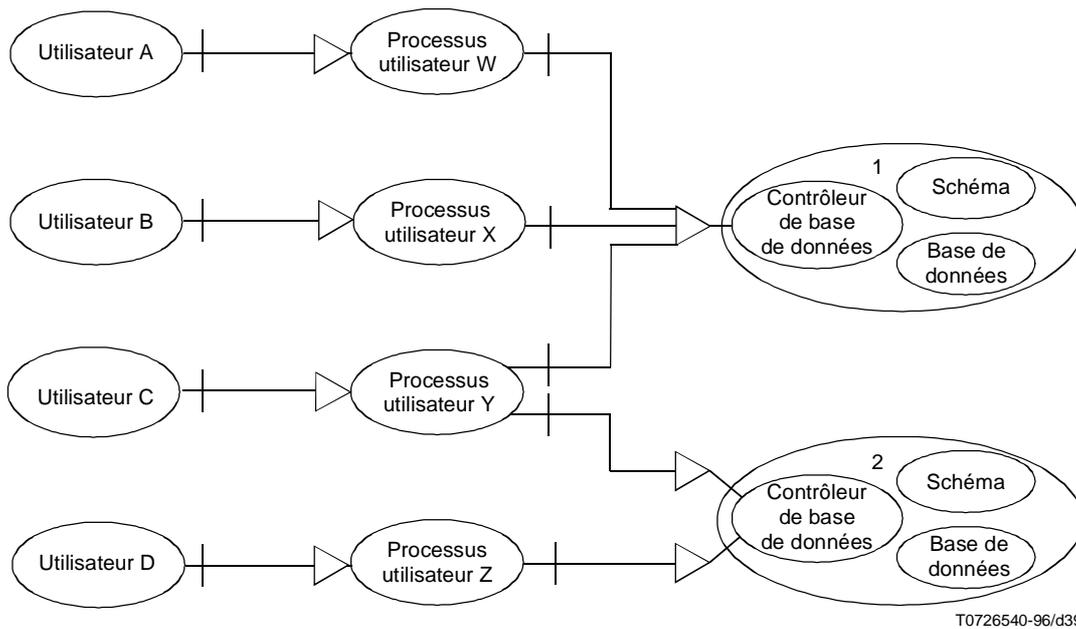


Figure 39 – Système de traitement de commandes – Spécification de traitement

Dans ce modèle de traitement, chaque objet utilisateur est d'un type correspondant soit à liaison client, soit à gestion d'entrepôt, soit à comptes. Chaque processus d'utilisateur fournit des services qui conviennent à son type d'utilisateur client, ce qui implique une partie, mais peu probablement la totalité, des objets du schéma d'information. Toutes les occurrences du schéma d'information, mais pas leurs opérations, sont contenues dans les objets de base de données. Pour les répartir entre des bases de données différentes, on peut appliquer des politiques diverses, comme, par exemple:

- l'affectation des objets en fonction du type, par exemple tous les objets clients et commandes dans une base de données et tous les objets entrepôts et produits dans une autre;
- l'affectation des objets en fonction d'une propriété donnée, par exemple une des bases de données conserve les instances de tous les types d'objets qui se rapportent à un entrepôt particulier, disons pour les appareillages électriques, alors qu'une autre base conserve les instances de tous les types relatifs à un autre entrepôt, disons pour les matériaux de construction;
- une combinaison de ces deux politiques, complétée de duplication de certains objets, afin de prendre en compte diverses catégories d'exigences de traitement. Une décomposition plus poussée des objets est aussi possible.

## **Annexe A**

### **Bibliographie**

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

- [Gay 95] V. GAY, P. LEYDEKKERS et R. HUIS IN 'T VELD: Specification of Multiparty Audio and Video Interaction Based on the Reference Model of Open Distributed Processing, *Computer Networks and ISDN Systems* – Special issue on RM-ODP, 1995.
- [OMG IDL] The Common Object Request Broker: Architecture and Specification, OMG Document Number 91.12.1, Draft edition, December 1991.
- [Rumbaugh 91] J. RUMBAUGH *et al.*: Object oriented modelling and design, Prentice Hall, 1991.

## SÉRIES DES RECOMMANDATIONS UIT-T

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