



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

**X.902**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

(11/95)

**RÉSEAUX POUR DONNÉES ET COMMUNICATION  
ENTRE SYSTÈMES OUVERTS  
TRAITEMENT RÉPARTI OUVERT**

---

**TECHNOLOGIES DE L'INFORMATION –  
TRAITEMENT RÉPARTI OUVERT – MODÈLE  
DE RÉFÉRENCE: FONDEMENTS**

**Recommandation UIT-T X.902**

(Antérieurement «Recommandation du CCITT»)

---

## 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. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie 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. Le texte de la Recommandation X.902 de l'UIT-T a été approuvé le 21 novembre 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10746-2.

---

### 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.

© UIT 1997

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.

RECOMMANDATIONS UIT-T DE LA SÉRIE X

**RÉSEAUX DE COMMUNICATION DE DONNÉES ET COMMUNICATION  
ENTRE SYSTÈMES OUVERTS**

(Février 1994)

**ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X**

Domaine	Recommandations
<b>RÉSEAUX PUBLICS POUR DONNÉES</b>	
Services et services complémentaires	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
Définition 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
Test de conformité	X.290-X.299
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
<b>SYSTÈMES DE MESSAGERIE</b>	X.400-X.499
<b>ANNUAIRE</b>	X.500-X.599
<b>RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES</b>	
Réseautage	X.600-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>	X.700-X.799
<b>SÉCURITÉ</b>	X.800-X.849
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
<b>TRAITEMENT OUVERT RÉPARTI</b>	X.900-X.999



## TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application.....	1
2	Références.....	1
	2.1 Recommandations   Normes internationales identiques.....	1
3	Définitions.....	1
	3.1 Définitions provenant d'autres Recommandations   Normes internationales.....	1
	3.2 Définitions de base.....	1
4	Abréviations.....	2
5	Catégories de concept.....	2
6	Concepts d'interprétation de base.....	3
7	Concepts linguistiques de base.....	3
8	Concepts de modélisation de base.....	3
9	Concepts de spécification.....	5
	9.1 composition.....	5
	9.3 décomposition.....	6
10	Concepts d'organisation.....	9
11	Propriétés des systèmes et des objets.....	10
	11.1 Transparences.....	10
	11.2 Concepts de politique.....	11
	11.3 Propriétés temporelles.....	12
12	Concepts de désignation.....	12
13	Concepts de comportement.....	12
	13.1 Structure d'activité.....	12
	13.2 Comportement contractuel.....	13
	13.3 Causalité.....	14
	13.4 Comportements d'établissement.....	14
	13.5 Sûreté de fonctionnement.....	15
14	Concepts de gestion.....	16
15	Approche ODP de la conformité.....	16
	15.1 Conformité aux normes ODP.....	16
	15.2 Tests et points de référence.....	16
	15.3 Classes de points de référence.....	16
	15.4 Changement de configuration.....	17
	15.5 Processus de test de conformité.....	17
	15.6 Résultat des tests.....	18
	15.7 Relation entre les points de référence.....	18

## Résumé

La présente Recommandation | Norme internationale définit les concepts et le cadre analytique servant à la description normalisée de systèmes (arbitraires) de traitement réparti. Elle introduit les principes de la conformité aux normes de traitement réparti ouvert (ODP) et la manière de les appliquer. Elle s'en tient à un niveau de détail suffisant pour les besoins de la Rec. X.903 | ISO/CEI 10746-3 et pour établir les prescriptions de nouvelles techniques de spécification.

## Introduction

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

Le Modèle de Référence pour le traitement réparti ouvert (RM-ODP) (*reference model of open distributed processing*), Rec. UIT-T X.901 à X.904 | ISO/CEI 10746, 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 description formelle pour la spécification de l'architecture.

Le Modèle de Référence ODP 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 champ 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 norme 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. Cette partie n'est pas normative;
- 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 introduit les principes de la conformité aux normes ODP et la manière dont ils s'appliquent. 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. Cette partie est normative;
- 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. Cette partie est normative;
- 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 présente Recommandation | Norme internationale (articles 8 et 9). La formalisation s'obtient en interprétant chaque concept à partir d'éléments des différentes techniques normalisées de description formelle. Cette partie est normative.

La présente Recommandation | Norme internationale ne comporte pas d'annexe.

## NORME INTERNATIONALE

## RECOMMANDATION UIT-T

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

### 1 Domaine d'application

La présente Recommandation UIT-T | Norme internationale traite des concepts nécessaires à la modélisation des systèmes ODP (voir les articles 5 à 14) ainsi que des principes de conformité aux systèmes ODP (voir l'article 15).

Les concepts définis dans les articles 5 à 15 sont utilisés dans le Modèle de Référence ODP pour définir:

- a) la structure de la famille des normes qui se réfèrent au Modèle de Référence;
- b) la structure des systèmes répartis revendiquant la compatibilité avec le Modèle de Référence (la configuration des systèmes);
- c) les concepts nécessaires pour l'utilisation combinée des diverses normes utilisées;
- d) les concepts de base qui sont utilisés dans les spécifications des divers composants qui constituent le système réparti ouvert.

### 2 Références

Les Recommandations et les 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 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.903 (1995) | ISO/CEI 10746-3:1996, *Technologies de l'information – Traitement réparti ouvert – Modèle de référence: Architecture.*

### 3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

#### 3.1 Définitions provenant d'autres Recommandations | Normes internationales

Il n'y a pas de définition provenant d'autres Recommandations | Normes internationales dans cette Recommandation | Norme internationale.

#### 3.2 Définitions de base

**3.2.1 traitement réparti:** Traitement de l'information dans lequel des composants déterminés peuvent être situés dans des lieux différents et au cours duquel les communications entre composants peuvent subir des délais ou échouer.

**3.2.2 normes ODP:** Le présent Modèle de Référence et les normes qui s'y conforment directement ou indirectement.

**3.2.3 traitement réparti ouvert:** Traitement réparti conçu pour être en conformité avec les normes ODP.

**3.2.4 système ODP:** Système (voir 6.5) conforme aux exigences des normes ODP.

**3.2.5 information:** Tout type de connaissance que peuvent s'échanger des utilisateurs dans un univers de discours, à propos de choses, faits, concepts, etc.

Bien que l'information ait nécessairement une forme de représentation qui la rend communicable, c'est l'interprétation de cette représentation (le sens) qui importe avant tout.

**3.2.6 données:** Formes de représentation de l'information que traitent les systèmes d'information et leurs utilisateurs.

**3.2.7 point de vue (sur un système):** Forme d'abstraction obtenue en utilisant un ensemble déterminé de concepts d'architecture et de règles de structuration, et permettant de se concentrer sur des préoccupations particulières liées à un système.

## 4 Abréviations

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

ODP	Traitement réparti ouvert ( <i>open distributed processing</i> )
OSI	Interconnexion des systèmes ouverts ( <i>open systems interconnection</i> )
PICS	Déclaration de conformité d'instance de protocole ( <i>protocol implementation conformance statement</i> )
PIXIT	Informations supplémentaires sur l'instance de protocole destinées au test ( <i>protocol implementation extra information for testing</i> )
RM-ODP	Modèle de référence du traitement réparti ouvert ( <i>reference model of open distributed processing</i> )
TP	Traitement transactionnel ( <i>transaction processing</i> )

## 5 Catégories de concept

Dans la présente Recommandation | Norme internationale on distingue les catégories de concept de modélisation suivantes:

- les concepts d'interprétation de base:* concepts destinés à l'interprétation des éléments de modélisation de tout langage de modélisation ODP. Ces concepts sont décrits dans l'article 6;
- les concepts linguistiques de base:* concepts liés aux langages; la grammaire de chaque langage utilisé dans l'Architecture ODP doit être décrite en utilisant ces concepts; ces concepts sont décrits dans l'article 7;
- les concepts de modélisation de base:* concepts servant à la construction de l'architecture ODP; les éléments de modélisation de tout langage doivent reposer sur ces concepts; ces concepts sont décrits dans l'article 8;
- les concepts de spécification:* concepts liés aux exigences des langages de spécification choisis et utilisés dans ODP. Ces concepts ne sont pas intrinsèques à la répartition et aux systèmes répartis, mais sont des exigences à prendre en compte dans ces langages de spécification; ces concepts sont décrits dans l'article 9;
- les concepts de structuration:* concepts de structuration qui ressortent de la considération des différents problèmes liés à la répartition et aux systèmes répartis. Ils peuvent ou non être directement pris en charge par des langages de spécification en adéquation avec le domaine concerné. L'utilisation des langages de spécification choisis doit permettre de spécifier des objets et des fonctions prenant directement en compte ces concepts; ces concepts sont décrits dans les articles 10 à 14;
- les concepts de conformité:* concepts nécessaires pour expliquer les notions de conformité aux normes ODP et de test de conformité; ces concepts sont définis dans l'article 15.

La Rec. UIT-T X.903 | ISO/CEI 10746-3 utilise les concepts définis dans la présente Recommandation | Norme internationale pour spécifier les caractéristiques d'un système réparti ouvert. Elle s'organise en un ensemble de langages de point de vue. Chaque langage de point de vue affine les concepts définis par la présente Recommandation | Norme internationale. Il n'est pas nécessaire que tous les langages points de vue adoptent les mêmes notations. Les différentes notations doivent être choisies en fonction des exigences du point de vue. Elles peuvent s'exprimer en langage naturel ou formel, être de nature textuelle ou graphique; toutefois, il sera nécessaire d'établir des correspondances entre les différents langages pour assurer une cohérence globale.

## 6 Concepts d'interprétation de base

Bien que l'Architecture ODP porte surtout sur la définition d'éléments formels, la sémantique du modèle architectural et de tous les langages de modélisation employés doit être décrite. Les concepts définis dans ce but sont essentiellement des métaconcepts, c'est-à-dire des concepts qui peuvent être utilisés dans toute forme d'activité de modélisation. Il n'est pas envisagé de définir formellement ces concepts ni de les utiliser comme base de définition formelle d'autres concepts.

Toute activité de modélisation identifie:

- a) les éléments de l'univers du discours;
- b) un ou plusieurs niveaux d'abstraction pertinents.

Les éléments de l'univers du discours sont des entités et des propositions.

**6.1 entité:** Tout élément concret ou abstrait, qui présente un intérêt. Alors que d'une manière générale le terme entité peut être utilisé pour faire référence à toute chose, son utilisation dans le contexte de la modélisation est réservée aux éléments modélisant l'univers du discours.

**6.2 proposition:** Fait ou état observable, impliquant une ou plusieurs entités, au sujet duquel il est possible d'affirmer ou de nier qu'il est vérifié pour ces entités.

**6.3 abstraction:** Processus consistant à supprimer un détail non pertinent pour créer un modèle simplifié; résultat de ce processus.

**6.4 atomicité:** Une entité est atomique à un niveau d'abstraction donné si on ne peut la subdiviser au niveau d'abstraction où on la considère.

Fixer un niveau donné d'abstraction peut impliquer l'identification des éléments atomiques.

**6.5 système:** Quelque chose qui présente un intérêt tant dans sa globalité que dans ses parties. Un système peut donc être considéré comme une entité. Un composant d'un système peut lui-même être un système, auquel cas il peut s'appeler sous-système.

NOTE – Pour les besoins de la modélisation, le concept de système doit être compris au sens de la théorie des systèmes. Le terme «système» peut se rapporter à un système de traitement de l'information mais peut aussi s'appliquer à un cadre plus général.

**6.6 architecture (d'un système):** Ensemble de règles destinées à définir la structure des systèmes, et les relations entre leurs différentes parties.

## 7 Concepts linguistiques de base

Un langage de modélisation utilisé dans le cadre de l'Architecture ODP, quels que soient ses concepts ou sa sémantique, sera exprimé dans une certaine syntaxe, qui peut inclure du texte linéaire ou des conventions graphiques. On suppose que tout langage approprié aura une grammaire définissant l'ensemble des symboles valides et les constructions linguistiques correctes du langage. Les concepts suivants permettent de relier la syntaxe de tout langage utilisé dans le cadre de l'Architecture ODP et les concepts d'interprétation.

**7.1 terme:** Structure linguistique qui peut être utilisée pour faire référence à une entité.

On peut faire référence à toute sorte d'entité, y compris à une modélisation d'une entité ou à une autre construction linguistique.

**7.2 phrase:** Structure linguistique contenant un ou plusieurs termes et prédicats; une phrase peut servir à exprimer une proposition relative aux entités que les termes désignent.

On considère qu'un prédicat appartenant à une phrase désigne une relation entre les entités désignées par les termes qu'elle lie.

## 8 Concepts de modélisation de base

L'interprétation détaillée des concepts définis dans cet article dépend du langage utilisé. Néanmoins, les énoncés généraux de ces concepts sont définis de façon indépendante des langages pour permettre la mise en relation des concepts définis dans les différents langages.

Les concepts de base se rapportent à l'existence et à l'activité: expression de ce qui existe, de l'endroit où cela se passe et de ce que cela fait.

**8.1 objet:** Modèle d'une entité. Un objet se caractérise par son comportement (voir 8.6) et, de manière duale, par son état (voir 8.7). Un objet est distinct de tout autre objet. Un objet est encapsulé, en ce sens qu'un changement de son état ne peut résulter que d'une action interne ou d'une interaction (voir 8.3) avec son environnement (voir 8.2).

L'interaction d'un objet avec son environnement intervient aux points d'interaction (voir 8.11).

Selon le point de vue, on met l'accent soit sur le comportement, soit sur l'état. Si on met l'accent sur le comportement, on dit, de manière informelle, qu'un objet exécute des fonctions et offre des services (on dit d'un objet qui met à disposition une fonction qu'il offre un service). Pour les besoins de la modélisation, ces fonctions et services sont spécifiés en termes de comportement de l'objet et de ses interfaces (voir 8.4). Un objet peut exécuter une ou plusieurs fonctions. Une fonction peut être réalisée par la coopération de plusieurs objets.

NOTES

1 Les concepts de service et de fonction sont utilisés d'une manière informelle pour caractériser l'intention d'une spécification de normalisation. Dans la famille des normes ODP, la fonction et le service sont formalisés par la spécification du comportement des objets et des interfaces utilisés. Un «service» est une abstraction particulière du comportement exprimant les garanties offertes par le fournisseur de service.

2 L'expression «utilisation d'une fonction» est une manière succincte de désigner l'interaction avec un objet qui fournit la fonction.

**8.2 environnement (d'un objet):** Partie du modèle qui ne fait pas partie de cet objet.

NOTE – Dans de nombreux langages de spécification, on peut estimer que l'environnement comprend au moins un objet susceptible de participer sans contrainte à toutes les interactions possibles (voir 8.3), représentant le processus d'observation.

**8.3 action:** Quelque chose qui se passe.

Toute action ayant un intérêt pour les besoins de la modélisation est associée à au moins un objet.

L'ensemble des actions associées à un objet est partitionné en **actions internes** et **interactions**. Une action interne se produit toujours sans la participation de l'environnement de l'objet. Une interaction se produit avec la participation de l'environnement de l'objet.

NOTES

1 «Action» signifie «occurrence d'action». Suivant le contexte, une spécification peut exprimer qu'une action s'est produite, est en cours ou peut se produire.

2 La granularité des actions est un choix de conception. Une action peut ne pas être instantanée. Des actions peuvent ainsi se chevaucher dans le temps.

3 Les interactions peuvent être définies en termes de relations de cause à effet entre les objets participants. Les concepts qui prennent cela en compte sont examinés en 13.3.

4 Un objet peut interagir avec lui-même (dans ce cas on considère qu'il joue au moins deux rôles). On peut considérer, dans ce contexte, qu'il fait partie de son propre environnement.

5 L'intervention de l'environnement représente ce qui est observable. On peut donc dire que les interactions sont observables alors que les actions internes ne le sont pas, du fait de l'encapsulation des objets.

**8.4 interface:** Abstraction du comportement d'un objet, qui se compose d'un sous-ensemble des interactions de cet objet, ainsi que d'un ensemble de contraintes portant sur les circonstances dans lesquelles ces interactions peuvent se produire.

Chaque interaction d'un objet appartient à une unique interface. Les interfaces d'un objet constituent donc une partition des interactions de cet objet.

NOTES

1 Une interface correspond à la partie du comportement d'un objet que l'on obtient en ne considérant que les interactions de cette interface et en cachant les autres interactions. Le fait de masquer les interactions des autres interfaces, de manière générale, introduit un non-déterminisme pour l'interface considérée.

2 L'expression «une interface entre objets» est utilisée pour désigner la liaison (voir 13.4.2) entre les interfaces des objets concernés.

**8.5 activité:** Graphe d'actions acyclique avec une seule racine, où l'occurrence de chaque action sur le graphe est rendue possible par l'occurrence de toutes les actions qui la précèdent immédiatement (c'est-à-dire par toutes les actions voisines qui sont plus proches du point d'origine).

**8.6 comportement (d'un objet):** Collection d'actions assortie d'un ensemble de contraintes portant sur les circonstances dans lesquelles ces actions peuvent se produire.

Les contraintes exprimables dépendent du langage de spécification utilisé. Elles peuvent par exemple comprendre des contraintes temps-réel, de séquençement, de non-déterminisme, ou de parallélisme.

Un comportement peut inclure des actions internes.

Les actions qui se produisent effectivement sont déterminées par l'environnement dans lequel se trouve l'objet.

#### NOTES

- 1 La composition (voir 9.1) d'une collection d'objets donne implicitement un objet équivalent représentant la composition. Le comportement de cet objet est souvent défini comme le comportement de la collection d'objets.
- 2 Action et activité sont des cas de comportement dégénéré.
- 3 En général, un comportement donné correspond à plusieurs séquences d'interactions.

**8.7 état (d'un objet):** A un instant donné dans le temps, condition d'un objet qui détermine l'ensemble de toutes les séquences d'actions auxquelles l'objet peut prendre part.

Le comportement impliquant, en règle générale, plusieurs séries d'actions possibles auxquelles l'objet pourrait prendre part, la connaissance de l'état ne permet pas nécessairement de prévoir la séquence des actions qui se produiront effectivement.

Les changements d'un état étant provoqués par des actions, un état est partiellement déterminé par les actions antérieures auxquelles l'objet a pris part.

Un objet étant encapsulé, son état ne peut être modifié directement par son environnement, mais seulement indirectement, à travers des interactions auxquelles l'objet prend part.

**8.8 communication:** Transmission d'informations entre deux objets ou plus, résultant d'une ou plusieurs interactions; ces interactions peuvent faire intervenir certains objets intermédiaires.

#### NOTES

- 1 Les communications peuvent être caractérisées en termes de relations de cause à effet entre les objets qui y participent. Les concepts afférents sont examinés en 13.3.
- 2 Chaque interaction est une instance d'une communication.

**8.9 position dans l'espace:** Intervalle d'espace de taille arbitraire où une action peut se produire.

**8.10 position dans le temps:** Intervalle de temps de taille arbitraire pendant lequel une action peut se produire.

#### NOTES

- 1 L'étendue de l'espace ou de l'intervalle de temps est choisie en fonction des exigences d'une spécification particulière ou des propriétés d'un langage de spécification donné. Une position définie dans une spécification donnée peut être divisée en temps ou en espace (ou les deux) dans une autre spécification. Dans une spécification donnée, une position dans le temps et dans l'espace est définie relativement à un système de coordonnées approprié.
- 2 Par extension, la position d'un objet correspond à l'union des positions des actions auxquelles l'objet prend part.

**8.11 point d'interaction:** Position où est présent un ensemble d'interfaces.

Selon le langage de spécification utilisé, étant donné une position dans le temps, on peut associer un point d'interaction à une position dans l'espace. Il peut y avoir plusieurs points d'interaction à la même position. Un point d'interaction peut être mobile.

## 9 Concepts de spécification

### 9.1 composition

- a) (d'objets): combinaison de deux objets ou plus, qui, à un autre niveau d'abstraction, résulte en un nouvel objet. Les objets combinés et la manière dont ils le sont déterminent les caractéristiques du nouvel objet. Le comportement d'un objet composite correspond à la composition des comportements des objets qui le composent;
- b) (de comportements): combinaison de deux comportements ou plus résultant en un nouveau comportement. Les comportements combinés et la manière dont ils le sont déterminent les caractéristiques du nouveau comportement.

#### NOTES

- 1 La composition séquentielle, la composition parallèle, l'entrelacement, le choix et le fait de masquer des actions sont des exemples de technique de combinaison. Ces définitions générales seront toujours employées dans un sens particulier, identifiant des moyens de combinaison spécifiques.
- 2 Dans certains cas, la composition de comportements peut engendrer un comportement dégénéré, par exemple l'interblocage; ceci est lié aux contraintes affectant les comportements d'origine.

**9.2 objet composite:** Objet résultant d'une composition.

### 9.3 décomposition

- a) (d'un objet): spécification d'un objet donné sous la forme d'une composition;
- b) (d'un comportement): spécification d'un comportement donné sous la forme d'une composition.

Composition et décomposition sont des termes duals et des activités de spécifications duales.

**9.4 compatibilité de comportement:** La compatibilité de comportement entre deux objets est déterminée relativement à un ensemble de critères (voir Notes). Un objet est dit compatible dans son comportement avec un deuxième objet, s'il peut remplacer le second sans que l'environnement ne puisse déceler de différence de comportement entre les deux objets.

Les critères de compatibilité de comportement imposent en général des contraintes sur le comportement possible de l'environnement. Si les critères sont tels que l'environnement se comporte comme un testeur de l'objet d'origine, c'est-à-dire si l'environnement définit le comportement minimal qui ne contraint pas le comportement de cet objet, on appelle extension la relation de compatibilité de comportement résultante.

Les critères de compatibilité de comportement peuvent autoriser la définition d'un objet de remplacement par modification d'un objet *a priori* incompatible. Une telle modification peut par exemple être réalisée en masquant des paramètres de certaines interactions. Ainsi, on peut faire ressembler une interaction du nouvel objet à une interaction de l'objet d'origine. Dans ce cas, on a une **compatibilité de comportement contrainte**. Si aucune modification n'est nécessaire, on a alors une **compatibilité de comportement naturelle**.

Le concept de compatibilité de comportement entre objets défini plus haut s'applique tout aussi bien à la compatibilité de comportement entre gabarits et types-gabarits.

La compatibilité de comportement est réflexive, mais pas forcément symétrique ou transitive (bien qu'elle puisse être l'une ou l'autre ou les deux).

#### NOTES

- 1 L'ensemble des critères de compatibilité de comportement dépend du langage utilisé et de la théorie de test appliquée.
- 2 La compatibilité de comportement (relativement à un ensemble de critères) peut être définie entre gabarits (voir 9.11) ou entre types-gabarits (voir 9.19):
  - a) si S et T sont des gabarits d'objet, on dit que S et T sont compatibles dans leur comportement si et seulement s'il y a compatibilité de comportement entre n'importe quelle instanciation de S et une quelconque instanciation de T (voir 9.13);
  - b) si U et V sont des types-gabarits d'objet, on dit qu'il y a compatibilité de comportement entre U et V si elle existe entre leurs gabarits.

**9.5 affinement:** L'affinement est le processus de transformation d'une spécification en une spécification plus détaillée. On peut dire de la nouvelle spécification qu'elle constitue un affinement de celle d'origine. En règle générale, les spécifications et leurs affinement ne coexistent pas dans la même description de système. Ce que l'on entend très précisément par spécification plus détaillée dépendra du langage de spécification choisi.

Pour chaque relation de compatibilité de comportement relativement à un ensemble de critères (voir 9.4), une technique de spécification permet de définir une relation d'affinement. Si un gabarit X affine un gabarit Y, il est possible de remplacer un objet instancié à partir de Y par un objet instancié à partir de X dans l'ensemble des environnements que détermine la définition choisie en termes de compatibilité de comportement. Les relations d'affinement ne sont pas forcément symétriques ou transitives.

**9.6 trace:** Enregistrement des interactions d'un objet, de son état initial à un quelconque autre état.

Une trace d'un objet est donc une séquence finie d'interactions. Le comportement détermine l'ensemble de toutes les traces possibles mais l'inverse n'est pas vrai. Une trace ne contient pas d'enregistrement des actions internes à un objet.

**9.7 type (d'un <X>):** Prédicat caractérisant une collection de <X>. Un <X> est du type ou satisfait au type si le prédicat s'applique à ce <X>. Une spécification définit parmi les termes qu'elle utilise quels sont ceux qui ont un type, c'est-à-dire ceux qui sont des <X>. Dans le Modèle de Référence ODP on a au minimum besoin des types pour les objets, les interfaces et les actions.

La notion de type classe implicitement les entités en catégories; certaines de ces catégories peuvent être intéressantes pour le spécificateur (voir le concept de classe en 9.8).

**9.8 classe (de <X>):** Ensemble de tous les <X> satisfaisant à un type (voir 9.7). Les éléments de l'ensemble sont désignés sous le terme de membres d'une classe.

## NOTES

- 1 Une classe peut ne pas avoir de membre.
- 2 La variation dans le temps de la taille de l'ensemble dépend de la définition du type.

**9.9 sous-type/supertype:** Un type A est un sous-type d'un type B, et B est un supertype de A, si chaque <X> satisfaisant au type A satisfait également au type B.

Les relations de sous-typage et de supertypage sont réflexives, transitives et antisymétriques.

**9.10 sous-classe/superclasse:** Une classe A est une sous-classe d'une autre classe B, et B est une superclasse de A, si tous les <X> satisfaisant à A satisfont aussi à B.

NOTE – Une sous-classe est par définition un sous-ensemble de chacune de ses superclasses.

**9.11 gabarit de <X>:** Spécification des caractéristiques communes à une collection de <X>, suffisamment détaillée pour permettre, en l'utilisant, l'instanciation d'un <X>. <X> peut être tout élément qui a un type (voir 9.7).

Un gabarit de <X> est une abstraction d'une collection de <X>.

Un gabarit peut spécifier des paramètres qui seront utilisés lors de l'instanciation.

La définition donnée ici est générique; la forme précise du gabarit dépendra de la technique de spécification utilisée. Les types de paramètre (quand ils sont utilisés) dépendront également de la technique de spécification utilisée.

Les gabarits peuvent se combiner en utilisant différentes méthodes de calcul. La forme précise de la combinaison des gabarits dépendra du langage de spécification utilisé.

**9.12 signature d'interface:** Ensemble des gabarits d'action associés aux interactions d'une interface.

Un objet peut avoir plusieurs interfaces de même signature.

**9.13 instanciation (d'un gabarit de <X>):** <X> produit à partir d'un gabarit de <X> donné et d'autres informations nécessaires. Ce <X> exhibe les caractéristiques spécifiées dans le gabarit de <X>. <X> peut être tout élément qui a un type (voir 9.7).

La définition donnée ici est générique; la nature précise de l'instanciation d'un gabarit de <X> dépendra du langage de spécification utilisé. Instancier un gabarit de <X> peut impliquer de donner une valeur de paramètre, ce qui peut déclencher l'instanciation d'autres gabarits de <X> ou la liaison d'interfaces existantes (voir 12.4).

## NOTES

- 1 L'instanciation d'un gabarit d'action a pour résultat une action qui se produit. L'expression «instanciation d'un gabarit d'action» est à proscrire. On lui préfère l'expression «occurrence d'une action».
- 2 Si <X> est un objet, il est instancié dans son état initial et peut ensuite participer à des interactions.
- 3 Des instanciations réalisées à partir de gabarits différents peuvent satisfaire au même type. Des instanciations réalisées à partir d'un même gabarit peuvent satisfaire à différents types.

**9.14 rôle:** Identificateur d'un comportement qui peut se présenter sous forme de paramètre dans un gabarit d'objet composite et qui est associé à l'un des objets composant l'objet composite.

La spécification d'un gabarit sous forme d'une composition de rôles permet d'expliquer le processus d'instanciation comme résultant de l'association d'un composant spécifique d'un objet composite avec chaque rôle. L'association d'un objet composant avec un rôle peut résulter de la valuation d'un paramètre.

**9.15 création (d'un <X>):** Instanciation d'un <X> qui résulte d'une action d'objets appartenant au modèle. <X> peut être toute chose instanciable, en particulier, un objet ou une interface.

Si <X> est une interface, elle peut être créée soit dans le cadre de la création d'un objet donné, ou comme interface supplémentaire pour un objet existant. Il en résulte que toute interface appartient à un objet.

**9.16 introduction (d'un <X>):** Instanciation d'un <X> qui ne résulte pas d'une action d'objets appartenant au modèle.

## NOTES

- 1 Un <X> peut être instancié par création ou bien par introduction, mais pas par les deux à la fois.
- 2 L'introduction ne peut s'appliquer ni aux interfaces, ni aux actions, puisqu'elles dépendent toujours d'objets déterminés

**9.17 suppression (d'un <X>):** Opération de destruction d'un <X> instancié. <X> peut être toute chose instanciable, en particulier, un objet ou une interface.

Une interface ne peut être supprimée que par l'objet qui la possède.

NOTE – La suppression d'une action n'a pas de sens: une action ne fait que se produire.

**9.18 instance (d'un type):** <X> satisfaisant à un type.

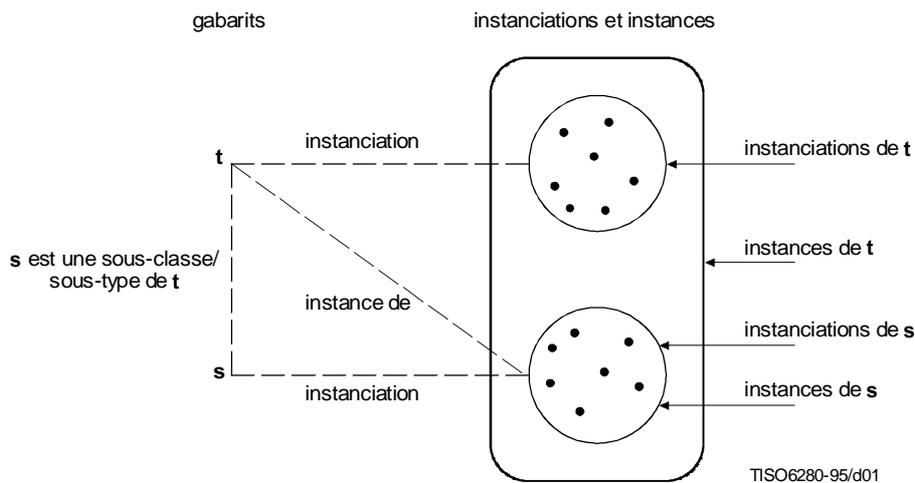
**9.19 type-gabarit (d'un <X>):** Prédicat défini dans un gabarit qui caractérise toutes les instanciations du gabarit et qui exprime les exigences que les instanciations du gabarit doivent remplir.

La relation de sous-typage entre types-gabarits d'objet ne coïncide pas forcément avec la relation de compatibilité de comportement. Les instances d'un type-gabarit n'ont pas forcément un comportement compatible avec des instanciations du gabarit associé. Les relations coïncident à condition:

- a) de considérer une relation de compatibilité de comportement transitive;
- b) d'avoir une compatibilité de comportement entre les sous-types-gabarits et leurs supertypes.

NOTES

- 1 Ce concept correspond à une notion de possibilité de substitution établie par conception.
- 2 La forme du prédicat qui exprime le type-gabarit dépend du langage de spécification utilisé.
- 3 De façon abrégée, on utilise le terme «instances d'un gabarit T» pour définir les instances du type-gabarit associé au gabarit T.
- 4 La Figure 1 illustre les relations existant entre certains des concepts: type-gabarit, classe de gabarit, etc. L'ensemble des instances du gabarit **t** contient à la fois l'ensemble des instanciations de **t** et les ensembles de toutes les instanciations des sous-types de **t**. Les ensembles d'instanciations de différents gabarits sont toujours disjoints.



**Figure 1 – Relations entre gabarits, instanciations et instances**

**9.20 classe de gabarit (d'un <X>):** Ensemble de tous les <X> satisfaisant à un type-gabarit de <X>, c'est-à-dire ensemble des <X> qui sont des instances du gabarit de <X>. <X> peut être toute chose qui a un type (voir 9.7).

Chaque gabarit définit une seule classe de gabarit; on peut donc désigner les instances du gabarit comme étant les instances de la classe de gabarit.

La notion de classe est utilisée pour désigner une classification générale de <X>. La notion de classe de gabarit est une notion plus restrictive. Les membres d'une classe de gabarit sont limités à ceux qui sont instanciés à partir du gabarit associé (ou de l'un de ses sous-types), c'est-à-dire les <X> satisfaisant au type-gabarit de <X>.

NOTE – Etant donné un type-gabarit, l'expression «la classe de gabarit associée au gabarit A est une sous-classe de la classe de gabarit associée au gabarit B» peut se réduire à l'expression «le gabarit A est une sous-classe du gabarit B» ou «le gabarit A est un sous-type du gabarit B».

**9.21 classe dérivée/classe de base:** Si un gabarit A est une modification incrémentale d'un gabarit B, alors la classe de gabarit CA des instances de A est une classe dérivée de la classe de gabarit CB des instances de B, et CB est une classe de base de CA.

Les critères permettant de déterminer si une modification donnée correspond à une modification incrémentale dépendent de métriques et de conventions non couvertes par la présente Recommandation | Norme internationale. Si les critères le permettent, une classe dérivée peut avoir plusieurs classes de base.

Une modification incrémentale liant les gabarits doit garantir que récurrence ou autoréférence dans le gabarit de la classe de base devienne récurrence ou autoréférence dans le gabarit de la classe dérivée.

D'une manière générale, une modification incrémentale peut entraîner une augmentation ou une modification des propriétés du gabarit de base pour obtenir le gabarit dérivé.

Les classes peuvent être organisées en une hiérarchie d'héritage selon les relations classe dérivée/classe de base. C'est ainsi qu'est interprété l'héritage dans le Modèle de Référence ODP. Si des classes peuvent avoir plusieurs classes de base, on dit que l'héritage est multiple. Si les critères de modification interdisent la suppression de propriétés provenant de la classe de base, on dit que l'héritage est strict.

Une classe peut être une sous-classe d'une seconde classe sans être une classe dérivée et être une classe dérivée sans être une sous-classe. La hiérarchie d'héritage (où les arcs indiquent la relation de classe dérivée) et la hiérarchie de type (où les arcs indiquent la relation de sous-typage ou de sous-classe) sont donc logiquement distinctes bien qu'elles puissent coïncider en totalité ou en partie.

**9.22 invariant:** Prédicat qui doit rester vrai pendant toute la durée de vie d'un ensemble d'objets.

**9.23 précondition:** Prédicat qui doit être vrai pour qu'une action ait lieu.

**9.24 postcondition:** Prédicat qui doit être vrai dès qu'une action a eu lieu.

## 10 Concepts d'organisation

**10.1 groupe de <X>:** Ensemble d'objets caractérisés par une relation <X> donnée. La relation <X> caractérise, soit une relation structurelle entre les objets, soit le comportement commun attendu des objets.

NOTE – Exemples de groupes spécialisés:

- a) *groupe d'adressage*: ensemble d'objets adressés de la même manière;
- b) *groupe de faute*: ensemble d'objets qui exhibent les mêmes types de faute. Par exemple, on peut supposer que si un ordinateur est défaillant, tous les objets exécutés sur cet ordinateur le seront également;
- c) *groupe de communication*: ensemble d'objets qui participent tous à la même séquence d'interactions avec leur environnement;
- d) *groupe de tolérance aux fautes et de duplication*: groupe de communication qui a pour but d'assurer un certain niveau de tolérance vis-à-vis de certaines fautes.

**10.2 configuration (d'objets):** Collection d'objets capables d'interagir à des interfaces. Une configuration détermine l'ensemble des objets qui sont impliqués dans chaque interaction.

La spécification de la configuration peut être statique ou dynamique. Une configuration dynamique implique l'utilisation de mécanismes de changement de configuration comme l'établissement ou la rupture de liaison entre objets (voir 13.4).

NOTE – Une configuration peut s'exprimer sous la forme d'une composition parallèle. La composition génère, à un autre niveau d'abstraction, un objet équivalent à la configuration.

**10.3 domaine de <X>:** Ensemble d'objets qui sont liés à un objet de contrôle par une relation caractéristique <X>.

A chaque domaine est associé un objet de contrôle.

Un objet de contrôle connaît l'identité de l'ensemble des objets qui constituent le domaine associé. Soit, il peut communiquer directement avec les objets contrôlés ou soit, on peut considérer que la communication a eu lieu à une époque antérieure (voir 10.5). En règle générale, un objet de contrôle n'appartient pas au domaine associé.

### NOTES

- 1 En termes d'entreprise, un objet de contrôle peut gérer diverses politiques liées au domaine.
- 2 Les domaines peuvent être disjoints ou se chevaucher.

- 3 Par définition, un domaine est un groupe, mais le contraire n'est pas vrai.
- 4 Exemples de domaines spécialisés:

Domaine	Classe de membres	Relation	Classe de contrôle
domaine de sécurité	objet de traitement	soumis à la politique établie par	objet autorité de sécurité
domaine de gestion	objet de gestion	soumis à la politique établie par	objet domaine de gestion
domaine d'adressage	objet adressé	adresse allouée par	objet autorité d'adressage
domaine de désignation	objet nommé	nom alloué par	objet autorité de désignation

**10.4 sous-domaine:** Domaine qui est un sous-ensemble d'un domaine donné.

**10.5 époque:** Période de temps pendant laquelle un objet présente un comportement particulier. Tout objet se trouve dans une seule époque à la fois, mais des objets interagissant peuvent se trouver à des époques différentes au moment de l'interaction.

Un changement d'époque peut correspondre à un changement de type d'un objet. Alternativement, un changement d'époque peut correspondre à une phase du comportement d'un objet de type constant.

Pour qu'un système réparti fonctionne correctement, les objets composant sa configuration doivent être cohérents. Quand un système évolue et traverse plusieurs époques, les objets en interaction qui constituent le système ne doivent jamais se trouver dans des époques telles que la mise en parallèle de leurs comportements soit impossible (c'est-à-dire conduite à une défaillance du système). Ce concept permet la formalisation de notions de version et d'extensibilité.

NOTE – Un langage de spécification peut avoir à exprimer:

- a) la façon dont on désigne une époque;
- b) la succession des époques et le fait de contraindre ou non tous les objets à traverser toutes les époques en séquence;
- c) les règles de dérivation de l'époque d'une composition à partir des époques de ses objets, en particulier pour des configurations et des systèmes complets;
- d) si l'identité de l'époque d'un objet fait nécessairement partie de l'état de cet objet;
- e) si les objets peuvent négocier sur la base de l'identité des époques où ils se trouvent;
- f) la relation entre époque et concepts de temps local et global.

**10.6 point de référence:** Point d'interaction défini dans une architecture susceptible d'être identifié comme point de conformité dans une spécification compatible avec cette architecture.

L'article 15 du Modèle de Référence ODP identifie des classes de point de référence pertinentes, ainsi que les relations existant entre modélisation et conformité.

**10.7 point de conformité:** Point de référence où observer un comportement à des fins de test de conformité.

## 11 Propriétés des systèmes et des objets

Le présent article décrit les propriétés de tout ou partie d'un système ODP.

### 11.1 Transparences

**11.1.1 transparence à la répartition:** Propriété qui permet de cacher à un utilisateur particulier le comportement potentiel de certaines parties du système.

NOTE – Les utilisateurs comprennent par exemple les utilisateurs finals, les développeurs d'application et de fonction.

## 11.2 Concepts de politique

**11.2.1 contrat:** Accord régissant une partie du comportement collectif d'un ensemble d'objets. Un contrat spécifie des obligations, permissions et interdictions qui s'appliquent aux objets concernés.

La spécification d'un contrat peut inclure:

- a) une spécification des différents rôles assumés par les objets impliqués dans le contrat et des interfaces associées à ces rôles;
- b) des attributs de qualité de service (voir 11.2.2);
- c) des indications de durée ou des périodes de validité;
- d) des indications de comportement invalidant le contrat;
- e) les conditions de sûreté et de vivacité.

### NOTES

1 Il n'est pas nécessaire que les objets impliqués dans le contrat soient liés hiérarchiquement, ils peuvent avoir une relation d'égal à égal. Les exigences liées à un contrat ne sont pas nécessairement applicables de la même manière à tous les objets concernés.

2 Un contrat peut s'appliquer en un point de référence donné d'un système. Dans ce cas, il spécifie le comportement auquel on peut s'attendre en ce point.

3 Un gabarit d'objet constitue un exemple simple de contrat. Un gabarit d'objet spécifie le comportement commun à une collection d'objets. A ce titre, il spécifie ce que l'environnement de ces objets peut supposer quant à leur comportement. A noter que des spécifications partielles de gabarit d'objet laissent le comportement des instances indéfini dans certaines circonstances (par exemple des interactions particulières); le contrat ne s'applique qu'au comportement spécifié.

**11.2.2 qualité de service:** Ensemble d'exigences de qualité relatives au comportement collectif d'un ou de plusieurs objets.

Les exigences de qualité de service peuvent se trouver dans un contrat ou être mesurées ou notifiées après un événement donné.

La qualité de service peut être paramétrée.

NOTE – La notion de qualité de service recouvre des caractéristiques telles que le débit de transfert des informations, le temps d'attente, la probabilité de défaillance de communication, la probabilité de défaillance d'un système, la probabilité de défaillance d'un système de stockage, etc.

**11.2.3 contrat d'environnement:** Contrat liant un objet et son environnement, qui inclut des contraintes de qualité de service, d'usage et de gestion.

Les contraintes de qualité de service comprennent:

- des contraintes temporelles (par exemple des dates limite);
- des contraintes de volume (par exemple des contraintes de débit);
- des contraintes de sûreté de fonctionnement couvrant des aspects liés à la disponibilité, la fiabilité, les facilités de maintenance, la sécurité (par exemple le temps moyen entre deux défaillances).

Les contraintes d'usage et de gestion comprennent:

- les contraintes de position (c'est-à-dire les positions dans le temps et dans l'espace qui sont choisies);
- les contraintes de transparence à la répartition (c'est-à-dire les transparences à la répartition sélectionnées).

Les contraintes de qualité de service peuvent impliquer des contraintes d'usage et de gestion. Certaines contraintes de qualité de service (par exemple la disponibilité) sont satisfaites grâce à la fourniture de transparences à la répartition (par exemple la duplication).

Les contraintes d'environnement peuvent décrire à la fois:

- des exigences concernant l'environnement d'un objet, pour garantir un comportement correct de l'objet;
- des contraintes sur le comportement d'un objet dans un environnement correct.

**11.2.4 obligation:** Prescription stipulant la nécessité d'un comportement particulier. Une obligation est remplie par l'occurrence du comportement prescrit.

**11.2.5 permission:** Prescription autorisant un comportement particulier. Une permission équivaut au fait qu'il n'est pas obligatoire que le comportement ne se produise pas.

**11.2.6 interdiction:** Prescription stipulant qu'un comportement particulier ne doit pas se manifester. Une interdiction équivaut au fait qu'il est obligatoire que le comportement ne se présente pas.

**11.2.7 politique:** Ensemble de règles liées à un objectif particulier. Une règle peut s'exprimer sous forme d'obligation, de permission ou d'interdiction.

NOTE – Les politiques ne sont pas toutes des contraintes. Certaines politiques représentent une attribution de pouvoirs.

## 11.3 Propriétés temporelles

**11.3.1 persistance:** Propriété d'un objet qui continue à exister à travers les changements de contexte contractuel (voir 13.2.3) ou d'époque.

**11.3.2 isochronicité:** Propriété d'une séquence d'actions où chaque paire d'actions adjacentes dans la séquence occupe des intervalles de temps adjacents, de même taille et uniques.

## 12 Concepts de désignation

**12.1 nom:** Terme qui, dans un contexte de désignation donné, désigne une entité.

**12.2 identificateur:** Nom non ambigu dans un contexte de désignation donné.

**12.3 espace de désignation:** Ensemble de termes pouvant être utilisés comme noms.

**12.4 contexte de désignation:** Relation entre un ensemble de noms et un ensemble d'entités. L'ensemble de noms appartient à un espace de désignation unique.

**12.5 action de désignation:** Action au cours de laquelle une entité donnée est associée à un terme d'un espace de désignation.

Toutes les actions de désignation sont relatives à un contexte de désignation.

**12.6 domaine de désignation:** Sous-ensemble d'un contexte de désignation tel que toutes les actions de désignation sont réalisées par l'objet de contrôle du domaine (l'objet autorisé de désignation).

NOTE – Un «domaine de désignation» est une instance du concept domaine de <X> (voir 10.3).

**12.7 graphe de désignation:** Graphe orienté où chaque sommet désigne un contexte de désignation et chaque arête une association entre:

- un nom apparaissant dans le contexte de désignation source;
- le contexte de désignation cible.

NOTE – L'existence d'une arête entre deux contextes de désignation sur un graphe de désignation signifie que le contexte de désignation cible peut être atteint (identifié) à partir du contexte de désignation source.

**12.8 résolution de nom:** Processus par lequel, un nom et un contexte de désignation initiaux étant donnés, on peut trouver une association entre un nom et l'entité désignée par le nom initial.

NOTE – L'opération de résolution de nom ne fournit pas nécessairement des informations suffisantes pour interagir avec l'entité désignée.

## 13 Concepts de comportement

### 13.1 Structure d'activité

**13.1.1 chaîne (d'actions):** Séquence d'actions dans le cadre d'une activité où, pour chaque paire d'actions adjacentes, l'occurrence de la première action est une condition nécessaire à l'occurrence de la seconde.

**13.1.2 fil d'exécution:** Chaîne d'actions dans laquelle un objet au moins participe à toutes les actions de la chaîne.

Un objet peut être associé à un seul fil ou à plusieurs fils d'exécution simultanément.

**13.1.3 action de jonction:** Action partagée entre deux chaînes ou plus et donnant lieu à une chaîne unique.

**13.1.4 action de division:** Action activant deux chaînes ou plus.

Il existe deux cas d'action de division, selon que les chaînes à diviser doivent se rejoindre ou non.

**13.1.5 action de branchement:** Action de division dans laquelle les chaînes activées doivent (sauf défaillance) se rejoindre entre elles; dans ce cas, les chaînes activées ne peuvent se joindre à d'autres chaînes et ne peuvent terminer séparément.

**13.1.6 action de génération:** Action de division dans laquelle les chaînes activées ne se joignent pas; dans ce cas, les chaînes activées peuvent interagir et se terminer séparément.

**13.1.7 action initiale:** Action qui n'a pas de prédécesseur, dans une activité donnée.

**13.1.8 sous-activité:** Sous-graphe d'une activité qui est lui-même une activité et remplit la condition suivante. Pour toute paire d'actions de jonction/branchement dans l'activité parente, les deux actions doivent être incluses dans le sous-graphe si l'une d'elles l'est.

## 13.2 Comportement contractuel

Les concepts introduits ici sont illustrés dans la Figure 2.

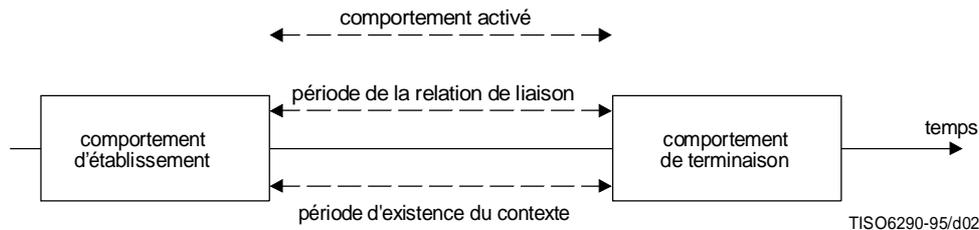


Figure 2 – Relation de liaison et concepts associés

**13.2.1 comportement d'établissement:** Comportement qui permet d'établir un contrat donné entre des objets donnés. Un comportement d'établissement peut être:

- a) explicite, c'est-à-dire résulter des interactions des objets qui participeront au contrat;
- b) implicite, c'est-à-dire être produit par un agent extérieur (par exemple un objet tiers ne participant pas au contrat), ou bien avoir été établi à une époque antérieure.

### NOTES

1 La négociation est un exemple particulier de comportement d'établissement, dans lequel des informations sont échangées dans le but de parvenir à une vision commune du comportement activé futur.

2 La publication est un exemple particulier de comportement d'établissement dans lequel les informations sont distribuées depuis un objet vers un certain nombre d'autres.

3 Un comportement d'établissement explicite doit inclure une instanciation du gabarit associé au contrat, ce qui peut suivre une éventuelle négociation/publication pour déterminer quel contrat doit être établi, quel gabarit instancié et avec quels paramètres.

**13.2.2 comportement activé:** Comportement caractérisant un ensemble d'objets rendu possible à la suite d'un comportement d'établissement.

Le comportement activé ne sera pas forcément le même pour tous les objets.

**13.2.3 contexte contractuel:** Connaissance de l'existence d'un contrat particulier et donc de l'exigence d'un comportement particulier pour un ensemble d'objets.

Un objet peut être simultanément dans un certain nombre de contextes contractuels, auquel cas le comportement est limité à l'intersection des comportements déterminés par chaque contexte contractuel.

NOTE – Dans l'OSI, le concept de contexte de présentation est un exemple de contexte et peut être établi au moment de l'établissement de la connexion ou ultérieurement.

**13.2.4 liaison:** Relation entre un ensemble d'objets résultant de la réalisation d'un comportement d'établissement; état consistant à avoir un contexte contractuel en commun.

Une liaison se caractérise par le comportement activé correspondant.

### NOTES

1 Exemples de liaisons résultant de différents comportements d'établissement:

- a) un dialogue (au sens d'OSI TP);
- b) une liaison (voir 13.4.2);
- c) une transaction répartie (au sens d'OSI TP);
- d) une (N)-connexion (au sens de l'OSI);
- e) une association entre des (N)-entités leur permettant de participer à une communication en mode sans connexion de niveau (N) (au sens de l'OSI);
- f) une relation entre les fichiers et les processus accédant aux fichiers.

2 Certains comportements peuvent être subordonnés à l'établissement de plusieurs liaisons associées. Par exemple, une transaction répartie peut dépendre à la fois de la liaison entre les utilisateurs de la transaction et de l'association assurant la prise en charge. La liaison entre les utilisateurs de la transaction (la transaction répartie) peut continuer à exister mais être inactive lorsque l'association est rompue.

3 Une liaison peut faire intervenir plus de deux objets. Les objets intervenant dans une liaison n'ont pas forcément tous des rôles équivalents. Il peut exister ainsi des liaisons pour la collecte ou la répartition d'informations. Le nombre de participants et leurs rôles sont déterminés par le contrat défini par la liaison.

4 Il y a une dualité entre l'acceptation d'un contexte contractuel et des obligations afférentes et le comportement activé. En pratique, des structures de contexte peuvent s'imbriquer arbitrairement et un comportement d'établissement à un niveau donné peut accepter un contrat donnant lieu à une liaison à un niveau inférieur.

**13.2.5 comportement de terminaison:** Comportement qui rompt une liaison et supprime le contexte et le contrat correspondants.

Un comportement de terminaison doit être explicitement identifié comme tel dans le contrat si le comportement d'établissement était explicite.

### 13.3 Causalité

L'identification de relations de causalité permet de classer les rôles des objets interagissant. Cet article donne un ensemble de rôles de base.

La **causalité** est une contrainte sur le comportement d'objets en interaction. La causalité est spécifiée dans la définition des classes (ou sous-classes) des objets en interaction ou dans l'affinement des gabarits correspondants.

**13.3.1 objet initiateur (dans le cadre d'une communication):** Objet cause d'une communication.

NOTE – Identifier un objet initiateur dans le cadre d'une communication donnée suppose l'interprétation des intentions liées à la communication.

**13.3.2 objet répondeur:** Objet participant à une communication mais qui n'est pas l'objet initiateur.

**13.3.3 objet producteur (dans le cadre d'une communication):** Objet qui est la source des informations véhiculées.

L'emploi du terme n'implique pas de mécanisme de communication spécifique.

**13.3.4 objet consommateur (dans le cadre d'une communication):** Objet qui est le collecteur des informations véhiculées.

L'emploi du terme n'implique pas de mécanisme de communication spécifique.

**13.3.5 objet client:** Objet demandant l'exécution d'une fonction par un autre objet.

**13.3.6 objet serveur:** Objet exécutant une fonction pour le compte d'un objet client.

Des relations client/serveur de différentes natures (ou d'un niveau d'abstraction différent) peuvent exister entre un objet et différentes compositions des objets avec lesquels il communique.

NOTE – Informellement, un serveur fournit un service demandé par un client.

### 13.4 Comportements d'établissement

**13.4.1 comportement de rattachement:** Comportement d'établissement entre deux ou plusieurs interfaces (et de ce fait entre leurs objets supports).

NOTE – «Rattacher» signifie «exécuter un comportement de rattachement».

**13.4.2 rattachement:** Contexte contractuel résultant d'un comportement d'établissement donné.

Le comportement d'établissement, le contexte contractuel et le comportement activé peuvent impliquer deux interfaces d'objet seulement ou plus.

Un objet qui initie un comportement d'établissement peut ou non participer au comportement activé consécutif.

Le comportement activé (et, par analogie, le contexte contractuel) peut être uniforme (c'est-à-dire que les objets participants ont le même rôle) ou non uniforme (c'est-à-dire que les objets participants ont des rôles différents, par exemple client et serveur).

Il n'y a pas nécessairement de correspondance entre un objet qui initie un comportement d'établissement et un rôle particulier dans les comportements activés non uniformes correspondants (par exemple dans un contexte contractuel client/serveur, chaque objet pourrait de façon valide avoir initié le comportement d'établissement).

**13.4.3 précondition de rattachement:** Ensemble de conditions nécessaires pour la réussite de l'exécution du comportement de rattachement.

Les objets exécutant le comportement de rattachement doivent posséder les identificateurs de toutes les interfaces impliquées dans le rattachement. Il peut exister des préconditions supplémentaires.

**13.4.4 comportement de détachement:** Comportement mettant fin au rattachement.

**13.4.5 courtage:** Interaction entre des objets au cours de laquelle des informations concernant des contrats nouveaux ou potentiels sont échangées par l'intermédiaire d'un objet tiers. Le courtage fait intervenir:

- a) *une opération d'exportation:* fourniture d'un identificateur d'une interface sensée répondre à certaines exigences (c'est-à-dire offrir un contrat potentiel);
- b) *une opération d'importation:* fourniture d'un identificateur d'une interface qui répond à des exigences données, permettant un comportement de liaison futur (c'est-à-dire l'établissement d'un contrat).

## 13.5 Sûreté de fonctionnement

**13.5.1 défaillance:** Violation d'un contrat.

### NOTES

1 Le comportement spécifié dans un contrat est, par définition, le «comportement correct». Une défaillance correspond donc à un écart par rapport à un comportement correct.

2 Les différentes formes de défaillance d'un objet sont appelées ses modes de défaillance. On peut distinguer plusieurs types de mode de défaillance:

- les défaillances arbitraires (non-conformité à la spécification – mode de défaillance le plus courant);
- les défaillances d'omission (quand des interactions attendues n'ont pas lieu);
- les crash (défaillances d'omission persistantes);
- les défaillances temporelles (incorrectes du fait de l'occurrence d'événements à des dates incorrectes).

3 Une défaillance peut être perçue de manière différente par des objets qui appartiennent à l'environnement de l'objet qui la subit. Une défaillance peut être cohérente si tous les objets de l'environnement en ont la même perception ou incohérente si les objets en ont des perceptions différentes.

**13.5.2 erreur:** Partie de l'état d'un objet susceptible d'entraîner des défaillances; manifestation d'une faute dans un objet (voir 13.5.3).

### NOTES

1 Qu'une erreur entraîne ou non une défaillance dépend de la décomposition d'un objet, de sa redondance interne et de son comportement. Une action corrective peut empêcher une erreur d'être à l'origine d'une défaillance.

2 Une erreur peut être latente (c'est-à-dire non reconnue comme telle) ou détectée. Une erreur peut disparaître avant d'avoir été détectée.

**13.5.3 défaut:** Situation pouvant occasionner des erreurs dans un objet.

### NOTES

1 Des défauts provoquant une erreur peuvent apparaître entre le moment où un objet est spécifié et le moment où il est détruit. Des défauts survenus dans une époque antérieure (par exemple des défauts de conception) peuvent n'entraîner une défaillance qu'à une époque ultérieure (par exemple le moment de l'exécution).

2 Un défaut est actif ou dormant. Un défaut est actif lorsqu'il provoque une erreur. La présence de défauts actifs n'est déterminée que par la détection des erreurs.

3 Les défauts peuvent être:

- accidentels (ils se produisent ou sont créés fortuitement) ou intentionnels (créés délibérément);
- physiques (dus à certains phénomènes physiques) ou d'origine humaine (résultant d'un comportement humain);
- internes (la partie d'un état d'objet pouvant provoquer une erreur) ou externes (résultant d'une interférence de l'environnement ou d'une interaction avec lui);
- permanents ou temporaires.

4 Les définitions des défauts, erreurs et défaillances impliquent, de manière récursive, des dépendances causales entre défauts, erreurs et défaillances:

- un défaut peut conduire à une erreur (il conduira à une erreur s'il devient actif);
- une erreur peut conduire à une défaillance du système (elle conduira à une défaillance à moins que le système ne puisse la traiter);
- une défaillance apparaît lorsqu'une erreur affecte le fonctionnement correct du service rendu par un système (ou un élément de système).

**13.5.4 stabilité:** Un objet est stable par rapport à un mode de défaillance donné s'il n'a aucune possibilité d'exhiber ce mode de défaillance.

## 14 Concepts de gestion

La gestion dans ODP prend en compte la gestion des systèmes dans leur intégralité, y compris la gestion des applications et la gestion des communications.

**14.1 gestion d'application:** Gestion des applications dans un système ODP. Certains aspects de la gestion des applications sont communs à toutes les applications; ils correspondent à la gestion appelée aspects génériques de gestion. Les autres aspects de la gestion, spécifiques à une application donnée, sont appelés aspects spécifiques de gestion.

**14.2 gestion de communication:** Gestion des objets qui prennent en charge la communication entre les objets dans un système ODP.

**14.3 informations de gestion:** Connaissances relatives à des objets qui sont pertinentes pour leur gestion.

**14.4 rôle d'objet géré:** Vue de l'interface de gestion d'un objet géré dans un système ODP.

NOTE – Lorsque l'objet fournit des services de communication OSI, la gestion OSI désigne l'interface de gestion sous le nom d'objet de gestion.

**14.5 rôle d'objet gérant:** Vue d'un objet effectuant des actions de gestion.

**14.6 notification:** Interaction activée par un objet remplissant un rôle d'objet géré.

## 15 Approche ODP de la conformité

### 15.1 Conformité aux normes ODP

La conformité relie une mise en œuvre à une norme. Toute proposition qui est vraie dans la spécification doit l'être dans sa mise en œuvre.

Une déclaration de conformité est une assertion qui identifie les points de conformité dans une spécification et le comportement attendu à ces points. Des déclarations de conformité n'apparaissent que dans des normes qui ont pour objectif de contraindre les caractéristiques d'une mise en œuvre réelle, de telle sorte qu'il existe la possibilité de tester cette mise en œuvre.

Le Modèle de Référence ODP identifie certains points de référence dans l'architecture comme susceptibles d'être désignés comme points de conformité dans les spécifications, c'est-à-dire des points où des tests de conformité pourront être effectués et qui devront donc être de ce fait accessibles. Toutefois, l'exigence selon laquelle un point de référence particulier est considéré comme un point de conformité doit figurer explicitement dans la déclaration de conformité de la spécification concernée.

Les exigences de cohérence nécessaire entre une norme de la famille ODP (le Modèle de Référence ODP, par exemple) et une autre sont établies au cours du processus de normalisation. Le respect de ces exigences est appelé **compatibilité**.

Si une spécification est compatible, directement ou indirectement, avec d'autres normes, les propositions qui sont exactes dans ces normes le sont également dans une mise en œuvre conforme de la spécification.

### 15.2 Tests et points de référence

La validité d'une déclaration de conformité dans une mise en œuvre ne peut être déterminée que par des tests et repose sur une mise en correspondance des termes de la spécification avec des aspects observables de la mise en œuvre.

A un niveau quelconque d'abstraction, un test est une série de stimuli et d'événements observables, se produisant aux points prescrits, dits points de référence, et seulement en ces points. Ces points de référence sont des interfaces accessibles. Un composant de système faisant l'objet d'une déclaration de conformité est considéré comme une boîte noire, susceptible d'être testée uniquement au niveau de ses liens externes. Ainsi, par exemple, la conformité aux spécifications de protocole OSI ne dépend pas d'une quelconque structure interne du système à tester.

### 15.3 Classes de points de référence

Un point de conformité est un point de référence où un test peut être réalisé sur un objet pour vérifier s'il répond à un ensemble de critères de conformité. Une déclaration de conformité doit identifier où se trouve le point de conformité et quels critères sont satisfaits en ce point. On définit quatre classes de points de référence où appliquer des tests de conformité.

**15.3.1 point de référence de programmation:** Point de référence où peut être mise en œuvre une interface de programmation afin de permettre l'accès à une fonction. Une déclaration de conformité à un point de référence de programmation est formulée en termes de compatibilité de comportement au sens où un objet peut être remplacé par un autre. Une interface de programmation est une interface qui est réalisée grâce à un couplage avec un langage de programmation.

NOTE – Par exemple, un point de référence de programmation peut être défini dans une norme sur les bases de données afin de prendre en charge un couplage avec un langage à un niveau d'abstraction donné.

**15.3.2 point de référence physique:** Point de référence où il y a interaction entre le système et le monde extérieur.

NOTES

1 Un point de référence physique peut être, par exemple, une interface homme-machine ou l'interface d'un robot avec le monde extérieur.

2 Les conditions de conformité à un point de référence physique, à une interface homme-machine, sont formulées en termes de forme d'information présentée et en termes de métaphore d'interaction et de dialogues avec un être humain.

3 On peut trouver un point de référence physique, par exemple, dans une norme d'infographie.

**15.3.3 point de référence d'interfonctionnement:** Point de référence où peut être mise en œuvre une interface permettant la communication entre deux ou plusieurs systèmes. Une condition de conformité à un point de référence d'interfonctionnement est formulée en termes d'échange d'information entre deux ou plusieurs systèmes. La conformité à un point de référence d'interfonctionnement implique l'interconnexion de points de référence.

NOTE – Par exemple, les normes OSI reposent sur l'interconnexion de points de référence d'interfonctionnement (le support physique).

**15.3.4 point de référence d'échange:** Point de référence où un support de stockage physique, externe, peut être introduit dans le système. Une condition de conformité à un point de référence d'échange est formulée en termes de comportement d'un support physique (méthodes et formats d'accès) de manière à pouvoir enregistrer l'information sur un système puis à la transférer physiquement, directement ou indirectement, dans le but de l'utiliser sur un autre système.

NOTE – Par exemple, certaines normes sur l'échange d'information reposent sur les points de référence d'échange.

## 15.4 Changement de configuration

Les tests de conformité peuvent avoir lieu à un seul point de référence, ou impliquer la mise en œuvre cohérente de plusieurs configurations à plusieurs points de référence. Cela peut entraîner des tests de conformité pour vérifier le respect des exigences suivantes:

- a) l'exigence qui impose à un composant d'être capable de fonctionner après une opération préparatoire destinée à l'adapter à l'environnement local;
- b) l'exigence qui impose à un composant de fonctionner en respectant sa spécification à un point de référence particulier et ce, depuis sa phase d'initialisation;
- c) l'exigence qui impose à un composant de continuer à fonctionner après qu'au cours d'une opération, il a été déplacé dans un environnement similaire.

Ces différentes formes d'exigence correspondent aux propriétés testées suivantes:

**15.4.1 portabilité:** Propriété résultant du fait que les points de référence d'un objet permettent de l'adapter à une variété de configuration.

NOTE – Si le point de référence est un point de référence de programmation, cela peut résulter en une portabilité du code source ou une portabilité du code exécutable. S'il s'agit d'un point de référence d'interfonctionnement, la portabilité correspondra à une portabilité d'équipement.

**15.4.2 capacité de migration:** Propriété résultant du fait de pouvoir changer la configuration dans laquelle est impliqué un objet en remplaçant un point de référence de l'objet considéré par un autre pendant que l'objet est en cours d'utilisation.

## 15.5 Processus de test de conformité

La conformité est un concept qui peut s'appliquer à tout niveau d'abstraction. Par exemple, on peut s'attendre à une conformité physique très détaillée pour une norme définissant des polices de caractères. Dans le cas des règles de structuration d'écran on aura par contre une conformité physique beaucoup plus abstraite.

La difficulté des tests augmente avec le niveau d'abstraction d'une spécification. Plus les propositions à établir d'une mise en œuvre donnée sont abstraites, plus il faut interpréter d'informations spécifiques à la mise en œuvre considérée. Il n'est pas évident que des tests directs de spécifications très abstraits soient possibles à un coût raisonnable en utilisant les techniques courantes ou à venir.

Le processus de test fait référence à une spécification. Pour être complète la spécification doit contenir:

- a) le comportement de l'objet en cours de normalisation et la façon donc ce comportement doit être réalisé;
- b) une liste des termes de base utilisés dans la spécification pour la rédaction des déclarations de comportement;
- c) une déclaration de conformité indiquant les points de conformité, le comportement qu'une mise en œuvre doit exhiber à ces points de conformité et quelles informations les réalisateurs doivent fournir (ceci correspond aux notions OSI de PICS et de PIXIT).

Il y a deux rôles dans le processus de test: le réalisateur et le testeur. Le réalisateur met en œuvre la spécification. Il doit fournir une déclaration qui donne la correspondance de tous les termes utilisés dans la spécification avec les objets ou événements du monde réel. Ainsi, il doit indiquer la correspondance entre interface et point de conformité et donner la représentation des interactions. Si la spécification est abstraite, la mise en relation de termes de base de la spécification avec le monde réel peut être complexe. Par exemple, dans une spécification de traitement (voir la Rec. UIT-T X.903 | ISO/CEI 10746-3), les termes de base correspondent à un ensemble d'interactions entre objets. Le réalisateur souhaitant être conforme à la spécification de traitement doit indiquer comment les interactions sont réalisées, soit en se référant à une spécification d'ingénierie, soit en fournissant une description détaillée d'un mécanisme non normalisé (dans ce dernier cas le champ d'application du système mis en œuvre est limité à l'ensemble des systèmes conformes aux mécanismes non normalisés).

Le testeur observe le système à tester. Les tests impliquent un certain comportement partagé entre le processus de test et le système à tester. Si ce comportement est globalement considéré comme causal, on distingue néanmoins toute une variété de types de tests:

- a) du test passif dans lequel tout comportement est à l'initiative du système à tester et enregistré par le testeur;
- b) au test actif dans lequel le comportement est à l'initiative du testeur et enregistré par lui.

La spécification du système à tester prend normalement la forme d'une interface, tout comme la spécification du testeur et des procédures de test. Lors du test, ces interfaces sont liées.

Le testeur doit interpréter ses observations en utilisant la mise en correspondance fournie par le réalisateur. Cela lui permet d'aboutir à des propositions relatives à des mises en œuvre dont la validité peut être ensuite vérifiée par rapport à la spécification.

## **15.6 Résultat des tests**

Le processus de test réussit si toutes les vérifications sont positives. Il peut toutefois échouer si:

- a) la spécification est incohérente ou incomplète, de sorte que les propositions relatives à la mise en œuvre ne peuvent être contrôlées (cela ne devrait pas se produire);
- b) la mise en correspondance fournie par le réalisateur est incomplète, si bien qu'elle est incohérente ou que les observations ne peuvent être reliées aux termes de la spécification; dans ce cas, le test est impossible;
- c) le comportement observé ne peut être interprété selon la mise en correspondance fournie par le réalisateur. Le comportement du système ne correspond pas à ce qui a été spécifié et le test est un échec;
- d) le comportement est interprété pour donner des termes exprimés dans la spécification, mais ils apparaissent de telle manière qu'ils aboutissent à des propositions qui sont fausses dans la spécification et le test est un échec.

## **15.7 Relation entre les points de référence**

Le flux d'information entre les composants d'un système modélisé peut traverser plusieurs points de référence. Par exemple, un système réparti peut impliquer des interactions de deux composants A et B, mais la communication entre les deux composants peut traverser tour à tour une interface de programmation, un point d'interconnexion et une autre interface de programmation.

Un affinement d'un même système peut montrer que des composants interconnectés peuvent avoir plus d'un composant sur l'axe de communication qui les relie.

Dans les deux cas, le test de conformité peut impliquer:

- a) le test du flux d'information à chacun de ces points de référence;
- b) le test de la cohérence entre les événements intervenant au niveau de couples de points de référence.

En général, tester le comportement correct d'une configuration d'objets exigera de tester si les déclarations relatives aux interfaces de communication sont vraies; cela impliquera également d'observer les autres interfaces de ces objets, de sorte que les déclarations relatives à leur composition puissent aussi être contrôlées.

La notion générale de conformité tient compte de la relation entre plusieurs points de conformité. La spécification liée à un point de conformité donné pouvant être exprimée à divers niveaux d'abstraction, les tests à un point de conformité donné impliqueront toujours une interprétation au niveau d'abstraction approprié. Ainsi, le test du comportement global nécessite des tests coordonnés à tous les points de conformité impliqués et l'utilisation de l'interprétation appropriée en chaque point.

En particulier, la conformité d'un gabarit à une interface de programmation donnée ne peut être établie qu'en considérant le couplage avec le langage de programmation dans lequel le gabarit a été écrit et la compatibilité de ce gabarit avec les spécifications du couplage qui doit lui-même être conforme à la spécification d'interface abstraite.

## Index

NOTE – A chaque terme correspond le numéro de paragraphe où celui-ci est défini.

- abstraction, 6.3
- action, 8.3
- action de branchement, 13.1.5
- action de désignation, 12.5
- action de division, 13.1.4
- action de génération, 13.1.6
- action de jonction, 13.1.3
- action initiale, 13.1.7
- action interne, 8.3
- activité, 8.5
- affinement, 9.5
- architecture (d'un système), 6.6
- atomicité, 6.4
- capacité de migration, 15.4.2
- chaîne (d'actions), 13.1.1
- classe (de <X>), 9.8
- classe de base, 9.21
- classe de gabarit (d'un <X>), 9.20
- classe dérivée, 9.21
- communication, 8.8
- compatibilité, 15.1
- compatibilité de comportement, 9.4
- compatibilité de comportement contrainte, 9.4
- compatibilité de comportement naturelle, 9.4
- comportement (d'un objet), 8.6
- comportement activé, 13.2.2
- comportement d'établissement, 13.2.1
- comportement de rattachement, 13.4.1
- comportement de terminaison, 13.2.5
- composition, 9.1
- configuration (d'objets), 10.2
- contexte contractuel, 13.2.3
- contexte de désignation, 12.4
- contrat, 11.2.1
- courtage, 13.4.5
- création (d'un <X>), 9.15
- décomposition, 9.3
- défaillance, 13.5.1
- défaut, 13.5.3
- détachement, 13.4.4
- domaine de <X>, 10.3
- domaine de désignation, 12.6
- données, 3.2.6
- entité, 6.1
- environnement (d'un objet), 8.2
- époque, 10.5
- erreur, 13.5.2
- espace de désignation, 12.3
- état (d'un objet), 8.7
- fil d'exécution, 13.1.2
- gabarit de <X>, 9.11
- gestion d'application, 14.1
- gestion de communication, 14.2
- graphe de désignation, 12.7
- groupe de <X>, 10.1
- identificateur, 12.2
- information, 3.2.5
- information de gestion, 14.3
- instance (d'un type), 9.18
- instanciation (d'un gabarit de <X>), 9.13
- interaction, 8.3
- interdiction, 11.2.5
- interface, 8.4
- introduction (d'un objet), 9.16
- invariant, 9.22
- isochronicité, 11.3.2
- liaison, 13.2.4
- nom, 12.1
- normes ODP, 3.2.2
- notification, 14.6
- objet, 8.1

objet client, 13.3.5  
 objet composite, 9.2  
 objet consommateur, 13.3.4  
 objet initiateur, 13.3.1  
 objet producteur, 13.3.3  
 objet répondeur, 13.3.2  
 objet serveur, 13.3.6  
 obligation, 11.2.3  
 permission, 11.2.4  
 persistance, 11.3.1  
 phrase, 7.2  
 point d'interaction, 8.11  
 point de conformité, 10.7  
 point de référence, 10.6  
 point de référence d'échange, 15.3.4  
 point de référence d'interfonctionnement, 15.3.3  
 point de référence de programmation, 15.3.1  
 point de référence physique, 15.3.2  
 point de vue (d'un système), 3.2.7  
 politique, 11.2.7  
 portabilité, 15.4.1  
 position dans l'espace, 8.9  
 position dans le temps, 8.10  
 postcondition, 9.24  
 précondition, 9.23  
 précondition de rattachement, 13.4.3  
 proposition, 6.2  
 qualité de service, 11.2.2  
 rattachement, 13.4.2  
 résolution de nom, 12.8  
 rôle, 9.14  
 rôle d'objet gérant, 14.5  
 rôle d'objet géré, 14.4  
 signature d'interface, 9.12  
 sous-activité, 13.1.8  
 sous-classe, 9.10  
 sous-domaine, 10.4  
 sous-type, 9.9  
 stabilité, 13.5.4  
 superclasse, 9.10  
 supertype, 9.9  
 suppression (d'un <X>), 9.17  
 système, 6.5  
 système ODP, 3.2.4  
 terme, 7.1  
 trace, 9.6  
 traitement réparti ouvert, 3.2.3  
 traitement réparti, 3.2.1  
 transparence à la répartition, 11.1.1  
 type (d'un <X>), 9.7  
 type-gabarit (d'un <X>), 9.19