



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**X.901**

(08/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN  
ENTRE SISTEMAS ABIERTOS

Procesamiento distribuido abierto

---

**Tecnología de la información – Procesamiento  
distribuido abierto – Modelo de referencia:  
Visión de conjunto**

Recomendación UIT-T X.901

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES DE LA SERIE X DEL UIT-T  
**REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS**

<b>REDES PÚBLICAS DE DATOS</b>	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
<b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
<b>INTERFUNCIONAMIENTO ENTRE REDES</b>	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.399
<b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>	<b>X.400–X.499</b>
<b>DIRECTORIO</b>	<b>X.500–X.599</b>
<b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b>	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
<b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
<b>SEGURIDAD</b>	<b>X.800–X.849</b>
<b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Cometimiento, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
<b>PROCESAMIENTO DISTRIBUIDO ABIERTO</b>	<b>X.900–X.999</b>

## **NORMA INTERNACIONAL 10746-1**

### **RECOMENDACIÓN UIT-T X.901**

# **TECNOLOGÍA DE LA INFORMACIÓN – PROCESAMIENTO DISTRIBUIDO ABIERTO – MODELO DE REFERENCIA: VISIÓN DE CONJUNTO**

## **Resumen**

Esta Recomendación | Norma Internacional es parte integrante del modelo de referencia de procesamiento distribuido abierto. Contiene una visión de conjunto motivada del procesamiento distribuido abierto (ODP, *open distributed processing*), que da el alcance, la justificación y la explicación de conceptos esenciales, y una descripción de la arquitectura ODP. Contiene material explicativo sobre la interpretación y aplicación del modelo de referencia por los usuarios, entre los que puede haber escritores de normas y arquitectos de sistemas ODP. Contiene también una agrupación en categorías de las áreas de normalización requeridas, expresadas en términos de los puntos de referencia para conformidad identificados en la Rec. UIT-T X.903 | ISO/CEI 10746-3.

## **Orígenes**

El texto de la Recomendación UIT-T X.901 se aprobó el 9 de agosto de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 10746-1.

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido/no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 1998

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

## ÍNDICE

		<i>Página</i>
1	Alcance y campo de aplicación .....	1
2	Referencias normativas .....	1
	2.1 Recomendaciones   Normas Internacionales idénticas .....	1
	2.2 Recomendaciones   Normas Internacionales de contenido técnico equivalente .....	2
	2.3 Normas Internacionales .....	2
3	Definiciones .....	2
	3.1 Definiciones en esta Recomendación   Norma Internacional .....	2
	3.2 Definiciones tomadas de otras Recomendaciones   Normas Internacionales .....	2
4	Abreviaturas .....	6
5	Convenios .....	7
6	Normalización del ODP .....	7
	6.1 Objetivos y motivación .....	7
	6.2 Realización .....	9
	6.2.1 Modelado de objeto .....	9
	6.2.2 Especificaciones de puntos de vista .....	9
	6.2.3 Transparencia de distribución .....	10
	6.2.4 Conformidad .....	10
	6.3 Normas .....	10
	6.3.1 Modelo de referencia .....	10
	6.3.2 Normas específicas .....	11
7	Fundamentos .....	11
	7.1 Conceptos básicos de modelado .....	12
	7.1.1 Objetos .....	12
	7.1.2 Interfaces y puntos de interacción .....	12
	7.1.3 Comportamiento y estado .....	12
	7.2 Conceptos de especificación .....	13
	7.2.1 Composición/descomposición .....	13
	7.2.2 Compatibilidad en comportamiento .....	13
	7.2.3 Tipo y clase .....	14
	7.2.4 Plantillas .....	14
	7.2.5 Roles (o papeles) .....	14
	7.2.6 Clases de base y clases derivadas .....	14
	7.3 Conceptos de estructuración .....	15
	7.3.1 Grupos y dominios .....	15
	7.3.2 Denominación .....	15
	7.3.3 Contrato .....	15
	7.3.4 Enlace y vinculación .....	16
8	Arquitectura .....	16
	8.1 Marco arquitectural .....	16
	8.1.1 Puntos de vista .....	17
	8.1.2 Transparencias de distribución .....	17
	8.2 Lenguaje de empresa .....	18
	8.3 Lenguaje de información .....	20
	8.4 Lenguaje computacional .....	21
	8.4.1 Interfaces computacionales .....	23
	8.4.2 Modelo de vinculación .....	23
	8.4.3 Tipificación y subtipificación para interfaces computacionales .....	25
	8.4.4 Portabilidad .....	26

	<i>Página</i>
8.5	Lenguaje de ingeniería..... 27
8.5.1	Conglomerados, cápsulas y nodos ..... 27
8.5.2	Canales..... 28
8.5.3	Referencias de interfaz..... 31
8.5.4	Vinculación..... 31
8.5.5	Establecimiento de canal..... 31
8.5.6	Interfaces de gestión ..... 32
8.5.7	Interceptores ..... 32
8.5.8	Puntos de conformidad ..... 34
8.6	Lenguaje de tecnología ..... 34
8.7	Consistencia entre los puntos de vista..... 35
8.7.1	Consistencia del punto de vista de la empresa con otros puntos de vista..... 36
8.7.2	Correspondencias entre especificaciones computacionales y de ingeniería ..... 37
8.8	Funciones ODP..... 39
8.8.1	Funciones de gestión..... 39
8.8.2	Funciones de coordinación ..... 40
8.8.3	Funciones de depositario..... 41
8.8.4	Funciones de seguridad..... 41
8.9	Transparencias de distribución ODP ..... 42
8.9.1	Transparencia de acceso ..... 42
8.9.2	Transparencia de fallo..... 42
8.9.3	Transparencia de ubicación..... 42
8.9.4	Transparencia de migración ..... 42
8.9.5	Transparencia de persistencia ..... 43
8.9.6	Transparencia de reubicación ..... 43
8.9.7	Transparencia de replicación ..... 43
8.9.8	Transparencia de transacción..... 43
9	Evaluación de la conformidad..... 43
9.1	Evaluación de la conformidad y proceso de desarrollo..... 43
9.2	Evaluación de conformidad: Relaciones relevantes ..... 44
9.3	Puntos de conformidad y conceptos conexos..... 44
9.4	Especificaciones de conformidad ODP..... 45
9.4.1	Nivel de abstracción..... 45
9.4.2	Utilización de múltiples puntos de referencia ..... 46
9.5	Implicaciones de la conformidad de los lenguajes de punto de vista ..... 46
9.6	Actividades de evaluación de la conformidad..... 47
10	Gestión de los sistemas ODP ..... 47
10.1	Dominios de gestión ..... 47
10.2	Política de gestión..... 48
10.3	Modelado de estructuras de gestión..... 48
11	Utilización de normas en sistemas ODP..... 48
11.1	Punto de vista de la empresa..... 48
11.1.1	Especificación de empresa ..... 48
11.1.2	Aplicación de normas ..... 50
11.2	Punto de vista de la información..... 50
11.2.1	Especificación de información..... 50
11.2.2	Aplicación de normas ..... 50
11.3	Punto de vista computacional ..... 51
11.3.1	Especificación computacional..... 51
11.3.2	Aplicación de normas ..... 52
11.4	Punto de vista de la ingeniería ..... 52
11.4.1	Especificación de ingeniería ..... 52
11.4.2	Aplicación de normas ..... 52
11.5	Punto de vista de la tecnología..... 53
11.5.1	Especificación de tecnología..... 53
11.5.2	Aplicación de normas ..... 54

	<i>Página</i>
12 Ejemplos de especificaciones ODP .....	54
12.1 Sistema de conferencias multimedios .....	55
12.1.1 Introducción .....	55
12.1.2 Especificación de empresa .....	56
12.1.3 Especificación de información .....	57
12.1.4 Especificación computacional.....	58
12.1.5 Especificación de ingeniería .....	61
12.1.6 Especificación de tecnología.....	62
12.2 Vinculación de trenes audio/vídeo de múltiples partes .....	62
12.2.1 Descripción general .....	63
12.2.2 Especificación de empresa .....	63
12.2.3 Especificación de información .....	65
12.2.4 Especificación computacional.....	67
12.2.5 Especificación de ingeniería .....	69
12.2.6 Especificación de tecnología.....	71
12.3 Ejemplo de gestión – Objeto métrico.....	71
12.3.1 Especificación de empresa .....	72
12.3.2 Especificación de información .....	73
12.3.3 Especificación computacional.....	74
12.4 Ejemplo de base de datos.....	75
12.4.1 Especificación de empresa .....	75
12.4.2 Especificación de información .....	76
12.4.3 Especificación computacional.....	76
Anexo A – Bibliografía .....	78

## Introducción

El rápido crecimiento del procesamiento distribuido ha creado la necesidad de un marco de coordinación para la normalización del procesamiento distribuido abierto (ODP, *open distributed processing*). Este modelo de referencia de ODP proporciona tal marco. Crea una arquitectura dentro de la cual se puede integrar un soporte de distribución, interfuncionamiento y portabilidad.

El modelo de referencia de procesamiento distribuido abierto Rec. UIT-T X.901 | ISO/CEI 10746-1 a Rec. UIT-T X.904 | ISO/CEI 10746-4, se basa en conceptos precisos utilizados en trabajos actualmente en curso para el desarrollo del procesamiento distribuido y, en la medida de lo posible, en el empleo de técnicas de descripción formal para la especificación de la arquitectura.

El modelo de referencia de procesamiento distribuido abierto (ISO/CEI 10746) está constituido por:

- Rec. UIT-T X.901 | ISO/CEI 10746-1: **Visión de conjunto:** contiene una visión de conjunto motivada del ODP, que da el alcance, la justificación y la explicación de conceptos esenciales, y una descripción de la arquitectura ODP. Contiene material explicativo sobre la interpretación y aplicación del RM-ODP por los usuarios, entre los que puede haber escritores de normas y arquitectos de sistemas ODP. Contiene también una agrupación en categorías de las áreas de normalización requeridas, expresadas en términos de los puntos de referencia para conformidad identificados en la Rec. UIT-T X.903 | ISO/CEI 10746-3. Estos textos comunes no son normativos.
- Rec. UIT-T X.902 | ISO/CEI 10746-2: **Fundamentos:** contiene la definición de los conceptos y marco analítico para la descripción normalizada de sistemas de procesamiento distribuido (arbitrarios). La exposición se hace solamente a un nivel de detalle suficiente para la aplicación de la Rec. UIT-T X.903 | ISO/CEI 10746-3 y el establecimiento de los requisitos que cumplirán las nuevas técnicas de especificación. Estos textos comunes son normativos.
- Rec. UIT-T X.903 | ISO/CEI 10746-3: **Arquitectura:** contiene la especificación de las características que debe tener un procesamiento distribuido para que sea abierto. Éstas son las constricciones a que deben ajustarse las normas ODP. Emplea las técnicas descriptivas de la Rec. UIT-T X.902 | ISO/CEI 10746-2. Estos textos comunes son normativos.
- Rec. UIT-T X.904 | ISO/CEI 10746-4: **Semántica arquitectural:** contiene una normalización de los conceptos de modelado ODP definidos en la Rec. UIT-T X.902 | ISO/CEI 10746-2, cláusulas 8 y 9. La normalización se consigue interpretando cada concepto en término de las construcciones de las diferentes técnicas de descripción formal normalizadas. Estos textos comunes son normativos.

Esta Recomendación | Norma Internacional contiene un anexo.

La cláusula 6 explica los beneficios comerciales de los sistemas distribuidos abiertos, así como la forma en que el modelo de referencia ODP (RM-ODP) y sus normas ODP asociadas permitirán a las compañías recibir estos beneficios. Esta cláusula indica las "promesas" de los bloques constructivos ODP de tipo enchufe activo ("plug-and-play") y las herramientas de integración de sistemas para sistemas distribuidos.

Las cláusulas 7 a 10 explican lo que es el RM-ODP, y sus funciones distribuidas. Estas cláusulas explican las razones por las que el RM-ODP soporta el desarrollo de los bloques constructivos de tipo enchufe activo y las herramientas de integración de sistemas para sistemas distribuidos.

La cláusula 11 muestra la manera en que, en la especificación ODP de un sistema, se puede hacer referencia a las normas y especificaciones ODP elaboradas por otros grupos. Estas relaciones son la clave de la aptitud del ODP para permitir la integración de diversas tecnologías.

La cláusula 12 presenta ejemplos que demuestran la utilización del RM-ODP y la utilización de los principios que le sirven de base para resolver problemas comerciales.

## NORMA INTERNACIONAL

## RECOMENDACIÓN UIT-T

## TECNOLOGÍA DE LA INFORMACIÓN – PROCESAMIENTO DISTRIBUIDO ABIERTO – MODELO DE REFERENCIA: VISIÓN DE CONJUNTO

### 1 Alcance y campo de aplicación

Esta Recomendación | Norma Internacional:

- presenta una introducción y la motivación para el procesamiento distribuido abierto (ODP);
- proporciona una visión de conjunto del modelo de referencia del procesamiento distribuido abierto (RM-ODP, *reference model of open distributed processing*) y una explicación de sus conceptos esenciales;
- da directrices sobre la aplicación del RM-ODP.

Esta Recomendación | Norma Internacional presenta la visión de conjunto y la explicación detallada y puede consultarse de diversas formas cuando se lean las normas:

- a) si el lector tiene la intención de leer solamente esta Recomendación | Norma Internacional, para comprender de manera general la importancia del ODP para su organización, deberá concentrarse en la cláusula 6;
- b) si el lector tiene la intención de estudiar la totalidad del RM-ODP, debe leer la cláusula 6 antes de pasar a la Rec. UIT-T X.902 | ISO/CEI 10746-2 y Rec. UIT-T X.903 | ISO/CEI 10746-3;
- c) durante la lectura de la Rec. UIT-T X.902 | ISO/CEI 10746-2 y Rec. UIT-T X.903 | ISO/CEI 10746-3, el lector podrá desear consultar las cláusulas 7 a 10, cuyas explicaciones facilitan la comprensión de los diversos conceptos definidos en estos textos comunes;
- d) una vez efectuada una primera lectura de la Rec. UIT-T X.902 | ISO/CEI 10746-2 y Rec. UIT-T X.903 | ISO/CEI 10746-3, el lector podrá pasar a las cláusulas 11 y 10, que explican la utilización de normas en especificaciones de sistemas ODP y presentan algunos ejemplos de la aplicación de los conceptos del ODP en la especificación de sistemas.

### 2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y las Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

#### 2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: El modelo básico.*
- Recomendación UIT-T X.207 (1993) | ISO/CEI 9545:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la capa de aplicación.*
- Recomendación UIT-T X.720 (1993) | ISO/CEI 10165-1:1993, *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Modelo de información de gestión.*
- Recomendación UIT-T X.902 (1995) | ISO/CEI 10746-2:1996, *Tecnología de la información – Procesamiento distribuido abierto – Modelo de referencia: Fundamentos.*

- Recomendación UIT-T X.903 (1995) | ISO/CEI 10746-3:1996, *Tecnología de la información – Procesamiento distribuido abierto – Modelo de referencia: Arquitectura.*
- Recomendación UIT-T X.904 (1997) | ISO/CEI 10746-4:1998, *Tecnología de la información – Procesamiento distribuido abierto – Modelo de referencia: Semántica arquitectural.*

## 2.2 Recomendaciones | Normas Internacionales de contenido técnico equivalente

- Recomendación UIT-T X.290 (1995), *Metodología y marco de las pruebas de conformidad de interconexión de sistemas abiertos de las Recomendaciones sobre los protocolos para aplicaciones del UIT-T – Conceptos generales.*
- ISO/CEI 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*

## 2.3 Normas Internacionales

- ISO/CEI 11578-2<sup>1)</sup>: *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC) – Part 2: Interface Definition Notation.*
- ISO/CEI TR 10000:1995, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework.*

## 3 Definiciones

### 3.1 Definiciones en esta Recomendación | Norma Internacional

No hay definiciones en esta Recomendación | Norma Internacional.

### 3.2 Definiciones tomadas de otras Recomendaciones | Normas Internacionales

Esta Recomendación | Norma Internacional utiliza los siguientes términos definidos en la Rec. UIT-T X.902 | ISO/CEI 10746-2:

- abstracción;
- acción;
- acción interna;
- actividad;
- arquitectura;
- atomicidad;
- avería;
- cadena (de acciones);
- calidad de servicio;
- clase de base;
- clase de plantilla;
- clase;
- clase derivada;
- comercio ("*trading*");
- compatibilidad en comportamiento;
- comportamiento de establecimiento;
- comportamiento de terminación;
- comportamiento (de un objeto);
- comportamiento de vinculación;
- composición;

---

<sup>1)</sup> Se publicará.

- comunicación;
- configuración (de objetos);
- contexto contractual;
- contrato de entorno;
- contrato;
- creación;
- cumplimiento ("*compliance*");
- datos;
- descomposición;
- desvinculación;
- dominio de denominación;
- enlace ("*liaison*");
- entidad;
- entorno (de un objeto);
- error;
- estado;
- fallo;
- firma de interfaz;
- hilo ("*thread*");
- identificador;
- información;
- información de gestión;
- instancia (ejemplar);
- instanciación;
- interacción;
- Interfaz;
- introducción (de un <X>);
- invariante;
- nombre;
- normas ODP;
- notificación;
- objeto;
- objeto de consumidor;
- objeto de servidor;
- objeto de productor;
- objeto de cliente;
- objeto compuesto;
- objeto iniciador;
- objeto respondedor;
- obligación;
- permiso;
- persistencia;
- plantilla de acción;
- política;
- portabilidad;
- prohibición;

## ISO/CEI 10746-1 : 1997 (S)

- punto de referencia programática;
- punto de vista;
- punto de referencia;
- punto de referencia de interfuncionamiento;
- punto de referencia perceptual;
- puntos de conformidad;
- refinamiento;
- resolución de nombre;
- rol (sinónimo: papel);
- sistema;
- sistema ODP;
- subclase;
- subtipo;
- supresión;
- tipo de plantilla;
- tipo;
- transparencia de distribución;
- ubicación en el espacio;
- vinculación.

Esta Recomendación | Norma Internacional utiliza los siguientes términos definidos en la Rec. UIT-T X.903 | ISO/CEI 10746-3:

- acción de vinculación compuesta;
- acciones de vinculación primitivas;
- anuncio;
- autoridad de seguridad;
- canal;
- cápsula;
- comunidad;
- desactivación;
- dominio de seguridad;
- esquema dinámico;
- esquema estático;
- esquema de replicación;
- esquema invariante;
- federación;
- firma de interfaz de tren(es);
- firma de interfaz de operación;
- firma de interfaz de señal(es);
- flujo;
- función ODP;
- gestor de cápsula;
- gestor de conglomerado ("*cluster manager*");
- información de control de acceso;
- interceptor (sinónimo: interceptador);
- interfaz de comunicación;
- interfaz de operación;

- interfaz de señal(es);
- interfaz de tren(es);
- interrogación;
- invocación;
- lenguaje <Punto de vista>;
- migración;
- nodo;
- núcleo;
- objetivo (o valor deseado, o (de) destino);
- objeto de ingeniería básico;
- objeto de vinculación;
- objeto de protocolo;
- ocultar;
- plantilla de conglomerado ("*cluster template*");
- política de seguridad;
- punto de vista de la ingeniería;
- punto de vista de la información;
- punto de vista computacional;
- punto de vista de la empresa;
- punto de comprobación;
- punto de vista de la tecnología;
- reactivación;
- recuperación;
- reubicador;
- señal;
- stub;
- terminación;
- transparencia de acceso;
- transparencia de fallo;
- transparencia de migración;
- transparencia de transacción;
- transparencia de reubicación;
- transparencia de replicación;
- transparencia de persistencia;
- transparencia de ubicación;
- validar;
- verificación por punto de comprobación ("*checkpointing*");
- vinculación explícita;
- vinculación implícita;
- vinculador ("*binder*").

Esta Recomendación | Norma Internacional utiliza los siguientes términos definidos en ISO/CEI 9646:

- enunciado de conformidad de implementación;
- información suplementaria de implementación para pruebas;
- punto de control y observación.

Esta Recomendación | Norma Internacional utiliza los siguientes términos definidos en Rec. UIT-T X.200 | ISO/CEI 7498-1:

- sintaxis abstracta;
- sintaxis de transferencia;
- sistema abierto.

## 4 Abreviaturas

En esta Recomendación | Norma Internacional se utilizan las siguientes siglas:

A-profile	Perfil de aplicación ( <i>application profile</i> )
ACID	Atomicidad consistencia aislamiento durabilidad ( <i>atomicity consistency isolation durability</i> )
AE(I)	Entidad de aplicación (invocación) [ <i>application entity (invocation)</i> ]
ALS	Estructura de la capa de aplicación ( <i>application layer structure</i> )
AP(I)	Proceso de aplicación (invocación) [ <i>application process (invocation)</i> ]
API	Interfaz de programa de aplicación ( <i>application program interface</i> )
ASO	Objeto de servicio de aplicación ( <i>application service object</i> )
BEO	Objeto de ingeniería básico ( <i>basic engineering object</i> )
CAD	Diseño asistido por computador ( <i>computer aided design</i> )
CD	Disco compacto ( <i>compact disk</i> )
CIM	Fabricación integrada al computador ( <i>computer integrated manufacturing</i> )
CMIP	Protocolo de información de gestión común ( <i>common management information protocol</i> )
CMIS	Servicio de información de gestión común ( <i>common management information service</i> )
DL	Lenguaje de definición ( <i>definition language</i> )
F-profile	Perfil de formato y presentación ( <i>format and presentation profile</i> )
FDT	Técnicas de descripción formal ( <i>formal description techniques</i> )
GUI	Interfaz de usuario gráfico ( <i>graphical user interface</i> )
HCI	Interfaz persona-computador ( <i>human computer interface</i> )
HDTV	Televisión de alta definición ( <i>high definition TV</i> )
ICS	Enunciado de conformidad de implementación ( <i>implementation conformance statement</i> )
IDL	Lenguaje de definición de interfaz ( <i>interface definition language</i> )
IT	Tecnología de la información ( <i>information technology</i> )
IXIT	Información suplementaria de implementación para pruebas ( <i>implementation extra information for testing</i> )
MIM	Modelo de información de gestión ( <i>management information model</i> )
MMC(S)	(Sistema de) Conferencias multimedia ( <i>multimedia conferencing (system)</i> )
ODP	Procesamiento distribuido abierto ( <i>open distributed processing</i> )
OMG	Grupo de gestión de objetos ( <i>object management group</i> )
OMT	Técnica de modelado de objetos ( <i>object modelling technique</i> )
OSE	Entorno de sistema abierto ( <i>open system environment</i> )
OSF	Fundamento de soporte lógico abierto (sinónimo: fundamento de software abierto) ( <i>open software foundation</i> )
OSI	Interconexión de sistemas abiertos ( <i>open systems interconnection</i> )
PCO	Punto de control y observación ( <i>point of control and observation</i> )

QOS	Calidad de servicio ( <i>quality of service</i> )
RDA	Acceso a base de datos distante ( <i>remote database access</i> )
RM-ODP	Modelo de referencia de procesamiento distribuido abierto ( <i>reference model of open distributed processing</i> )
RPC	Llamada de procedimiento a distancia ( <i>remote procedure call</i> )
T-profile	Perfil de transferencia ( <i>transfer profile</i> )
TINA	Arquitectura de funcionamiento en red para información de telecomunicación ( <i>telecommunication information networking architecture</i> )
ULA	Arquitectura de capas superiores ( <i>upper layers architecture</i> )

## 5 Convenios

Los siguientes convenios son específicos de esta Recomendación | Norma Internacional:

- 1) En las cláusulas 7 y 8, la primera vez que se utilizan los términos formales tomados de la Rec. UIT-T X.902 | ISO/CEI 10746-2 y Rec. UIT-T X.903 | ISO/CEI 10746-3 se escriben en cursiva.
- 2) Para los ejemplos en la cláusula 12 se han utilizado los convenios de dibujo OMT definidos en [Rumbaugh 91].
- 3) En los diagramas:
  - Los objetos se representan como óvalos o círculos.
  - Un símbolo "⊥" que sobresale de un objeto representa una interfaz.

## 6 Normalización del ODP

### 6.1 Objetivos y motivación

El objetivo de la normalización del ODP es la elaboración de normas que permitan obtener los beneficios que ofrece la distribución de los servicios de procesamiento de la información que habrán de realizarse en un entorno de recursos de tecnología de la información (recursos IT) heterogéneos y múltiples dominios organizacionales. Estas normas imponen constricciones a la especificación de sistemas y la provisión de una infraestructura de sistema que tenga en cuenta las dificultades inherentes en el diseño y la programación de sistemas distribuidos.

Los sistemas distribuidos son importantes porque existe una necesidad creciente de interconectar sistemas de procesamiento de la información. Esta necesidad surge como consecuencia de tendencias organizacionales tales como la reducción del tamaño, que exige un intercambio de información tanto entre grupos en el interior de una organización como entre organizaciones cooperantes. Los avances tecnológicos han hecho posible responder a estas tendencias dando una importancia creciente a las redes de servicios de información y a las estaciones de trabajo personales, y permitiendo la construcción de aplicaciones distribuidas a través de vastas configuraciones de sistemas interconectados.

Con el fin de gestionar y explotar un sistema de distribución (por ejemplo, aprovechando su potencial de disponibilidad, calidad de funcionamiento, seguridad de funcionamiento y optimización de costos), las organizaciones deben tratar varias características esenciales de la distribución de sistemas:

- **Ubicación en lugares distantes:** Los componentes de un sistema distribuido pueden estar ubicados en distintos puntos en el espacio; las interacciones deben ser locales o distantes.
- **Concurrencia:** Todo componente de un sistema distribuido puede ejecutar acciones en paralelo con cualquier otro componente.
- **Ausencia del estado global:** El estado global de un sistema distribuido no puede determinarse con precisión.
- **Fallos parciales:** Todo componente de un sistema distribuido puede fallar independientemente de cualquier otro componente.
- **Asincronía:** Las actividades de comunicación y procesamiento no son cadenciadas por un reloj global único. No puede suponerse que en un sistema distribuido se produzcan cambios conexos en un mismo instante.

- **Heterogeneidad:** No está garantizado que los componentes de un sistema distribuido estén contruidos sobre la base de la misma tecnología y el conjunto de las diversas tecnologías seguramente cambiará con el transcurso del tiempo. La heterogeneidad aparece en muchos lugares: el soporte material, los sistemas operativos, redes y protocolos de comunicación, lenguajes de programación, aplicaciones, etc.
- **Autonomía:** Un sistema distribuido puede estar esparcido entre varias autoridades autónomas de gestión y control, sin que haya un punto único de control. El grado de autonomía especifica la medida en que los recursos de procesamiento y dispositivos asociados (impresoras, dispositivos de almacenamiento, de visualización gráfica, de audio, etc.) están bajo el control de distintas entidades organizacionales.
- **Evolución:** Durante su vida de servicio, un sistema distribuido generalmente tiene que sufrir muchos cambios motivados por el progreso técnico y que permiten obtener un mejor rendimiento a mejor precio, así como por decisiones estratégicas sobre nuevos objetivos, y por nuevos tipos de aplicación.
- **Movilidad:** Las fuentes de información, los nodos de procesamiento y los usuarios pueden ser físicamente móviles. Los programas y los datos pueden también ser desplazados entre los nodos, por ejemplo para estar en consonancia con la movilidad física o para optimizar la calidad de funcionamiento.

La construcción de tales sistemas no es fácil. Requiere una arquitectura y, dado que una solución técnica única no satisfará todos las exigencias, dicha arquitectura deberá ser flexible. Además, puesto que un solo vendedor no tendrá respuesta para todas las cuestiones, es esencial que la arquitectura y toda función necesaria para implementarla, estén definidas en un conjunto de normas, de modo que varios vendedores puedan colaborar en la provisión de sistemas distribuidos. Estas normas permitirán construir sistemas que:

- Sean **abiertos**, es decir, que proporcionen portabilidad (ejecución de componentes en diversos nodos de procesamiento, sin modificación), e interfuncionamiento (habrá interacciones significativas entre los componentes, que podrían residir en sistemas diferentes).
- Estén **integrados**, es decir, que incorporen diversos sistemas y recursos formando un todo, sin que para ello se necesiten costosos desarrollos especializados. Esto puede entrañar sistemas con arquitecturas diferentes, y recursos diferentes con calidades de funcionamiento diferentes. La integración ayuda a hacer frente a la heterogeneidad.
- Sean **flexibles**, es decir, capaces de evolucionar y de adaptarse a la existencia y operación continuada de sistemas heredados. Un sistema distribuido abierto debe ser capaz de hacer frente a los cambios que se producen en la fase de ejecución, por ejemplo, debe poder reconfigurarse dinámicamente para adaptarse a circunstancias cambiantes. La flexibilidad ayuda a hacer frente a la movilidad.
- Sean **modulares**, es decir, que permitan a partes de un sistema ser autónomas, al mismo tiempo que están interrelacionadas. La modularidad es la base para la flexibilidad.
- Puedan estar **federados**, es decir, que permitan que un sistema se combine con sistemas de dominios administrativos o técnicos diferentes para alcanzar un objetivo único.
- Sean **gestionables**, es decir, permitan supervisar, controlar y gestionar los recursos de un sistema para las políticas de configuración, calidad de servicio y contabilidad.
- Satisfagan las necesidades de **calidad de servicio**, que abarcan, por ejemplo, la provisión de tempestividad (*timeliness*), disponibilidad y fiabilidad en el contexto de recursos e interacciones distantes, junto con la provisión de una tolerancia a las averías que permita a la parte restante de un sistema distribuido seguir funcionando en caso de fallo de una de sus partes. La provisión de una tolerancia a las averías (y de la seguridad de funcionamiento en general) es necesaria en los grandes sistemas distribuidos, en los que es improbable que todas las partes del sistema estén siempre operacionales en todo momento.
- Sean **seguros**, es decir, que garanticen que las facilidades y los datos de un sistema están protegidos contra un acceso no autorizado. Los requisitos de seguridad son más difíciles de cumplir cuando las interacciones se producen entre puntos distantes, y las partes del sistema y de los usuarios del sistema son móviles.

- Ofrezcan **transparencia**, que enmascare, ocultándolos con respecto a las aplicaciones, los detalles y las diferencias en los mecanismos utilizados para resolver los problemas causados por la distribución. Éste es un requisito central que surge de la necesidad de facilitar la construcción de aplicaciones distribuidas. Entre los aspectos de la distribución que deben ser enmascarados (total o parcialmente) se encuentran: la heterogeneidad del soporte lógico y del soporte físico, la ubicación y movilidad de los componentes, y los mecanismos para lograr el nivel requerido de calidad de servicio en presencia de fallos (por ejemplo, replicación, migración, verificación por puntos de comprobación, etc.).

## 6.2 Realización

La normalización del procesamiento distribuido abierto tiene cuatro elementos fundamentales:

- un método de modelado de objeto para la especificación de sistema;
- la especificación de un sistema en términos de especificaciones de punto de vista distintas, pero interrelacionadas;
- la definición de una infraestructura de sistema que proporcione transparencias de distribución para aplicaciones de sistema;
- un marco para evaluar la conformidad del sistema.

### 6.2.1 Modelado de objeto

El modelado de objeto proporciona una formalización de prácticas de diseño bien conocidas, de abstracción y encapsulación. La abstracción permite separar la descripción de una funcionalidad de sistema como distinta de los detalles de la implementación del sistema. La encapsulación permite ocultar la heterogeneidad, el lugar en que se produjo un fallo, la implementación de seguridad, así como ocultar el mecanismo de provisión de servicio con respecto al usuario del servicio.

Los conceptos de modelado de objeto abarcan:

- Conceptos de modelado básico – Que proporcionan definiciones rigurosas de un conjunto mínimo de conceptos (acción, objeto, interacción e interfaz) que forman la base de las descripciones de sistemas ODP y son aplicables en todos los puntos de vista.
- Conceptos de especificación – Que tratan nociones como las de tipo y clase y son necesarios para razonar sobre las especificaciones y las relaciones entre las especificaciones, proporcionar herramientas generales para el diseño, y establecer requisitos de los lenguajes de especificación.
- Conceptos de estructuración – Basados en los conceptos de modelado básico y los conceptos de especificación para tratar estructuras recurrentes en sistemas distribuidos, y abarcar temas tales como las políticas, denominación, comportamiento, seguridad de funcionamiento, y comunicación.

### 6.2.2 Especificaciones de puntos de vista

Un punto de vista (sobre un sistema) es una abstracción que produce una especificación del sistema total en relación con un conjunto particular de temas. Se han elegido cinco puntos de vista de modo que sean a la vez sencillos y concretos, y que abarcan todos los dominios del diseño de arquitecturas. Estos cinco puntos de vista son:

- punto de vista de la empresa, que se relaciona con la finalidad, alcance y políticas que rigen las actividades del sistema especificado dentro de la organización de que forma parte;
- punto de vista de la información, que se relaciona con las clases de información tratadas por el sistema y las constricciones impuestas a la utilización e interpretación de esa información;
- el punto de vista computacional, que se relaciona con la descomposición funcional del sistema en un conjunto de objetos que interactúan en interfaces, permitiendo así la distribución del sistema;
- punto de vista de la ingeniería, que se relaciona con la infraestructura requerida para soportar la distribución del sistema;
- punto de vista de la tecnología, que se relaciona con la elección de la tecnología que soportará la distribución del sistema.

Para cada punto de vista hay un lenguaje de punto de vista asociado que puede utilizarse para expresar una especificación del sistema, desde ese punto de vista. Los conceptos de modelado de objeto proporcionan una base común para los lenguajes de punto de vista y hacen posible identificar relaciones entre las diferentes especificaciones de punto de vista y determinar correspondencias entre las representaciones del sistema en diferentes puntos de vista.

### 6.2.3 Transparencia de distribución

Las transparencias de distribución permiten ocultar las complejidades relacionadas con la distribución de un sistema, con respecto a las aplicaciones en que dichas complejidades son irrelevantes para esa finalidad. Por ejemplo:

- la transparencia de acceso enmascara las diferencias de representación de datos y mecanismos de invocación para servicios entre sistemas;
- la transparencia de ubicación enmascara la necesidad que tiene una aplicación de información sobre la ubicación con el fin de invocar un servicio;
- la transparencia de reubicación enmascara la reubicación de un sistema con respecto a las aplicaciones que lo utilizan;
- la transparencia de replicación enmascara el hecho de que se pueden proporcionar múltiples copias de un mismo servicio para proporcionar fiabilidad y disponibilidad.

Las normas ODP definen funciones y estructuras para realizar transparencias de distribución. Sin embargo, para cada transparencia hay que realizar negociaciones entre la calidad de funcionamiento y el costo, y sólo transparencias escogidas serán relevantes en muchos casos. Así, un sistema ODP conforme no está obligado a soportar todas las transparencias, pero las transparencias que soporte deberá implementarlas de acuerdo con las normas pertinentes.

### 6.2.4 Conformidad

Las características básicas de heterogeneidad y evolución implican que diferentes partes de un sistema distribuido puedan adquirirse separadamente, de diferentes vendedores. Por ello es muy importante que los comportamientos de las diferentes partes de un sistema estén claramente definidos, y que sea posible asignar responsabilidad por cualquier fallo, para cumplir las especificaciones del sistema.

El marco definido para regir la evaluación de la conformidad trata estas cuestiones. Abarca:

- identificación de los puntos de conformidad dentro del conjunto de especificaciones de punto de vista en los que pueden efectuarse observaciones de la conformidad;
- definición de clases de punto de conformidad;
- especificación de la naturaleza de los enunciados de conformidad que habrán de hacerse en cada punto de vista y la relación entre los mismos.

## 6.3 Normas

### 6.3.1 Modelo de referencia

El modelo de referencia del procesamiento distribuido abierto (RM-ODP) proporciona el marco genérico para la normalización del procesamiento distribuido abierto. Consta de dos partes principales:

- Rec. UIT-T X.902 | ISO/CEI 10746-2: **Fundamentos**, que define los conceptos y el marco analítico para la descripción de sistemas de procesamiento distribuido, incluyendo un marco general para la evaluación de conformidad.
- Rec. UIT-T X.903 | ISO/CEI 10746-3: **Arquitectura**, que define la forma de especificar sistemas ODP y la infraestructura que proporciona transparencias de distribución.

La Rec. UIT-T X.904 | ISO/CEI 10746-4: **Semántica arquitectural** complementa estas dos partes principales al proporcionar una interpretación formal de los conceptos de modelado y lenguajes de punto de vista en términos de técnicas de descripción formal existentes.

El RM-ODP es genérico, es decir, independiente de, e igualmente aplicable a, dominios de aplicación arbitrarios que utilizan o requieren la tecnología de los sistemas distribuidos. Para algunos dominios de aplicación específicos será necesario refinar y especializar el RM-ODP para que satisfaga necesidades particulares, de lo cual resultarán:

- modelos de referencia específicos que abarcan tipos individuales de empresa, utilizan conceptos y funciones comunes que figuran en el RM-ODP, y definen detalles conceptuales y funciones específicas adicionales, por ejemplo la arquitectura de funcionamiento en red de la información de telecomunicación (TINA, *telecommunication information networking architecture*);
- normas para la realización de funciones específicas necesarias para aplicaciones particulares y posiblemente identificadas en un modelo de referencia específico, por ejemplo interfaces para la conexión de llamadas telefónicas.

Por el hecho de ser genérico el RM-ODP permite también integrar diferentes tecnologías de sistemas distribuidos con el fin de obtener soluciones de sistema técnicas, eficaces con relación al costo, para las exigencias de los negocios. En particular, en el caso de las arquitecturas publicadas por la fundación de software abierto (OSF, *open software foundation*) y el grupo de gestión de objetos (OMG, *object management group*) para explicar cómo las funciones que ellos especifican para el soporte de sistemas distribuidos concuerdan unas con otras, el método del ODP añade valor al tratar tales cuestiones como una gestión de federación, transferencia y sistema, y definiendo un marco, de grano fino, de puntos de referencia para el soporte de la integración de funciones provenientes de diferentes fuentes.

### 6.3.2 Normas específicas

En el marco global proporcionado por el RM-ODP se han identificado cuatro categorías de normas:

- marcos arquitecturales adicionales, que complementan el RM-ODP en áreas específicas tales como denominación, seguridad y evaluación de la conformidad;
- normas de notación, que definen notaciones para expresar especificaciones de diferentes aspectos de la integración y distribución de sistemas, y reglas para relacionar especificaciones diferentes;
- normas de componentes, que definen una función ODP única o un conjunto de funciones ODP estrechamente interrelacionadas, que posiblemente pueda implementarse como una plataforma única de soporte físico (*hardware*) y lógico (*software*);
- normas de composición de componentes que definen la utilización coordinada de varios componentes para alcanzar algún objetivo del sistema en su conjunto, por ejemplo como una provisión de una transparencia específica.

NOTA – Algunas normas pueden especificar los componentes y su composición (lo que permite implementar directamente una facilidad útil). Otras normas pueden formar la base para varias normas de composición de componentes, por ejemplo, en normas de composición de componentes se haría referencia a una norma de reubicador ODP para la provisión de transparencia de ubicación o migración.

El RM-ODP proporciona un marco para normas de componentes y normas de composición de componentes para funciones ODP, que permite aplicar varios métodos diferentes para su realización. Esta flexibilidad es necesaria si el marco debe tener un tiempo de vida razonable, e incorporar nuevos desarrollos a medida que estos maduren. Así, una norma única o un conjunto de normas especifican una solución particular para el cumplimiento de algún requisito ODP, tomando todas las decisiones específicas necesarias para hacer posible la implementación de productos abiertos, y puede haber varias de esas normas, correspondientes a diferentes opciones de diseño. Con el transcurso del tiempo se incorporarán nuevas tecnologías, lo que conducirá a nuevas generaciones de normas dentro de un marco ODP único.

## 7 Fundamentos

La Rec. UIT-T X.902 | ISO/CEI 10746-2 define un conjunto de conceptos de modelado que proporcionan el fundamento para expresar la *arquitectura de sistemas ODP* definidos en la Rec. UIT-T X.903 | ISO/CEI 10746-3. Estos conceptos se han agrupado en tres categorías:

- conceptos de modelado básico que introducen un modelo general basado en objetos. En general, un sistema ODP puede describirse como una colección de *objetos* conexos interactuantes;
- conceptos de especificación que no son intrínsecos a sistemas distribuidos pero que permiten a quien los utilice describir y razonar sobre especificaciones de sistemas ODP. Estos conceptos imponen requisitos a todo lenguaje de especificación que se utilice para la especificación de un sistema ODP y, por ser esencialmente independientes del lenguaje, pueden aplicarse en cualquier lenguaje de especificación o en cualquier lenguaje de programación determinados;
- conceptos de estructuración, que comprenden la organización, las propiedades de sistemas y objetos, *política, denominación, comportamiento* y gestión, que corresponden a nociones y estructuras que son aplicables de manera general en el diseño y la descripción de sistemas distribuidos.

La Rec. UIT-T X.902 | ISO/CEI 10746-2 proporciona también un marco general para facilitar la comprensión de la *conformidad* en sistemas ODP, expresado en términos del modelo de objeto general. Este marco se analiza en la cláusula 9.

## 7.1 Conceptos básicos de modelado

### 7.1.1 Objetos

Las especificaciones de sistemas ODP se expresan en términos de objetos. Un objeto es una representación de una *entidad* en el mundo real. Contiene *información* y ofrece servicios. Un sistema se compone de objetos interactuantes. Un objeto se caracteriza por aquello que lo distingue de otros objetos así como por la *encapsulación*, *abstracción* y comportamiento.

La encapsulación es la propiedad por la cual la información contenida en un objeto es accesible solamente mediante *interacciones* en las *interfaces* soportadas por el objeto. Puesto que los objetos están encapsulados, las interacciones no producen efectos marginales ocultos. Esto es, una interacción con un objeto no puede afectar al *estado* de otro objeto sin que se produzca alguna interacción secundaria con ese objeto. Así, todo cambio en el estado de un objeto sólo puede producirse como resultado de una *acción interna* del objeto o como resultado de una interacción del objeto con su *entorno*.

La abstracción implica que los detalles internos de un objeto están ocultos con respecto a otros objetos y es de capital importancia para el tratamiento de la heterogeneidad, pues permite implementar servicios diferentes de maneras diferentes, utilizando mecanismos y tecnologías diferentes, lo que hace posible la *portabilidad* y la interoperabilidad.

La abstracción proporciona también una buena separación entre los objetos, lo que permite reemplazarlos o modificarlos sin cambiar su entorno, a condición de que continúen soportando los servicios esperados por su entorno (es decir, que sean retrocompatibles). Este método para conseguir la extensibilidad es esencial en los entornos distribuidos grandes y heterogéneos, los cuales, por su propia naturaleza, se encuentran en constante evolución. Un modelo de objeto proporciona modularidad y la aptitud para componer nuevos módulos a partir de módulos existentes: estas capacidades son importantes para la construcción de sistemas flexibles y promueven la reutilización para acrecentar la productividad.

El modelo de objeto ODP es general y parte de un número mínimo de supuestos. Por ejemplo:

- los objetos pueden tener una granularidad arbitraria (es decir, pueden ser tan grandes como una red telefónica o tan pequeños como un número entero);
- los objetos pueden tener comportamientos (encapsulados) arbitrarios, y tener un nivel arbitrario de paralelismo interno;
- las interacciones entre los objetos no están constreñidas y pueden incluir, por ejemplo, interacciones asíncronas e interacciones síncronas multidireccionales.

### 7.1.2 Interfaces y puntos de interacción

Los objetos sólo pueden interactuar en interfaces, donde una interfaz representa una parte del comportamiento del objeto relacionado con un subconjunto particular de sus interacciones posibles. Cada interfaz está identificada por un conjunto de interacciones en las que el objeto puede participar. Obsérvese que estas interacciones no se producen necesariamente con otros objetos: un objeto puede interactuar consigo mismo. Una importante característica del concepto de objeto en el RM-ODP es que un objeto puede tener varias interfaces. Los motivos para considerar múltiples interfaces son separación funcional y la distribución. La separación funcional puede entenderse, por ejemplo, en el contexto de gestión de sistemas, en el que las interacciones de gestión y las interacciones que no son de gestión normalmente se separan en interfaces diferentes. Cuando se tratan los objetos distribuidos, la separación es también necesaria donde las interfaces constituyen puntos de acceso al objeto que están situados en diferentes lugares en el espacio.

Como una consecuencia de definir varias interfaces para un objeto, las interacciones en una de las interfaces pueden ser afectadas por las interacciones en otras interfaces, y no tienen necesariamente que ser determinadas aisladamente.

Una interfaz existe en un *punto de interacción* que, en cualquier instante, está asociado con algún punto en el espacio. Pueden existir varias interfaces en un punto de interacción dado y el punto de interacción puede ser móvil. El significado de los puntos de interacción en el espacio y en el tiempo y la manera de expresarlos dependen del lenguaje en que se expresa la especificación.

### 7.1.3 Comportamiento y estado

Un comportamiento de un objeto es una colección de *acciones* en las que el objeto puede participar, junto con un conjunto de constricciones impuestas sobre el momento en que esas acciones pueden producirse. El modelo de objeto no constriñe la forma o la naturaleza del comportamiento del objeto. Las acciones pueden ser interacciones del objeto con su entorno o acciones internas del objeto.

Los conceptos de estado y comportamiento están interrelacionados. El estado de un objeto es la condición del objeto en un instante dado que determina las secuencias futuras potenciales de acciones en que el objeto puede participar. Al mismo tiempo, las acciones producen cambios de estado y, por tanto, el estado actual de un objeto está determinado en parte por su comportamiento pasado. Desde luego, las acciones que efectivamente realizará el objeto no están totalmente determinadas por su estado presente; dependerán también de las acciones en las que el entorno esté preparado para participar.

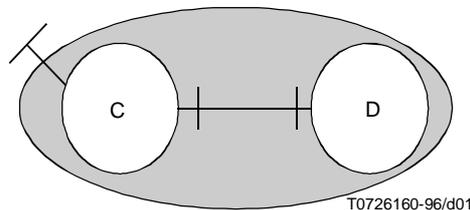
## 7.2 Conceptos de especificación

### 7.2.1 Composición/descomposición

La *composición* y la *descomposición* pueden utilizarse para organizar la especificación de un sistema distribuido como un conjunto de especificaciones, cada una de las cuales trata un nivel diferente de abstracción. Permiten descomponer la especificación de un sistema distribuido complejo en especificaciones de varios objetos más simples que también pueden ser descompuestos en un nivel de abstracción más bajo.

Los procesos de composición y descomposición proporcionan una especificación jerárquica de una aplicación distribuida. En esta composición jerárquica, las *clases* de objeto en los niveles más altos se ensamblan a partir de *configuraciones* de clases de objetos componentes en niveles más bajos. Por tanto, una composición es un poderoso concepto de modelado ya que permite tratar un subsistema como un objeto único de un nivel más alto.

A título de ejemplo, el diagrama de la figura 1 muestra la composición de dos objetos, C y D. Para componer C con D, el comportamiento de C debe definirse de modo que interactúe debidamente con D. La interfaz entre C y D puede ser, sencillamente, una interacción única (por ejemplo, pasar información de C a D) o puede ser un comportamiento más complejo (como una secuencia de interacciones en las que C invoca una operación que retorna un resultado). Si C y D se componen para formar un objeto, las interacciones entre C y D quedan ocultas y se convierten en acciones internas del *objeto compuesto*. La interfaz que aparece a la izquierda representa una interfaz del objeto compuesto.



**Figura 1 – Composición de objeto**

La composición de objetos produce la composición de estados y comportamientos, por lo que es posible hablar de un comportamiento compuesto y de un estado compuesto.

La jerarquía de la composición es ortogonal con la jerarquía de *subclase* tratada en 7.2.3, y no debe confundirse con ella. En general, no hay relación de jerarquía de subclase entre las clases de los objetos componentes y las clases del objeto compuesto. En efecto, existen ejemplos en los que el objeto compuesto pertenece a una subclase de la clase a la que pertenece el objeto compuesto (por ejemplo cuando los objetos representan servicios de comunicaciones, y el objeto compuesto añade valor a uno de sus componentes).

### 7.2.2 Compatibilidad en comportamiento

Se dice que un objeto es compatible en comportamiento con otro objeto en un entorno si el primer objeto puede reemplazar al segundo sin que el entorno pueda detectar alguna diferencia. Toda interpretación particular de la *compatibilidad en comportamiento* impondrá constricciones al comportamiento autorizado del entorno. Un método usual consiste en suponer que el entorno se comporta como un probador con respecto al objeto original. Es decir, el entorno debe ser capaz de realizar plenamente el comportamiento original, pero no debe poder hacer más que eso. Estos supuestos son esenciales si la compatibilidad en comportamiento ha de tomarse en consideración en el contexto de algún entorno desconocido.

### 7.2.3 Tipo y clase

Un *tipo* es un predicado (es decir, una propiedad o conjunto de propiedades) de una colección de cosas (objetos, interfaces, etc.). Por ejemplo "es rojo" es un tipo. Decimos que algo satisface a un tipo, o es de un tipo, si el predicado es cierto para esa cosa. Las cosas pueden ser disímiles y, a pesar de eso, ser del mismo tipo; sólo tienen que poseer las propiedades prescritas por el tipo. Por ejemplo, una determinada bandera, una determinada casa de ladrillos y un determinado automóvil deportivo pueden, todos ellos, ser rojos.

Los tipos clasifican implícitamente las cosas en conjuntos conocidos como clases, siendo una clase la colección de cosas con las propiedades prescritas por un tipo.

La noción de tipo es muy general y puede ser especializada de diversas maneras. Es útil en cualquier contexto en que sea necesario hablar y razonar sobre propiedades de cosas, y verificar propiedades de cosas (por ejemplo para *comercio* ("*trading*"), para *vinculación*).

Los conceptos de tipo y clase producen las naturales jerarquías de clase/subclase y tipo/subtipo. La distinción entre clase/subclase corresponde a la distinción intuitiva en la teoría de conjuntos entre conjuntos y subconjuntos. Una clase es una subclase de otra únicamente si la primera es un subconjunto de la segunda. Un tipo es un subtipo de otro si los predicados del primer tipo implican los predicados del segundo tipo.

La formación de subclases y la formación de subtipos están relacionadas. Para cada tipo hay una clase asociada (que, naturalmente, puede estar vacía). Por tanto, si tenemos dos tipos T1 y T2, tienen que existir las clases asociadas C1 y C2. T1 es un subtipo de T2 precisamente cuando C1 sea una subclase de C2.

### 7.2.4 Plantillas

Una *plantilla* describe una colección de cosas (objetos, interfaces etc.) lo suficientemente en detalle para que se pueda crear un ejemplar (dícese una instancia) de una nueva cosa a partir de ella.

Cuando una plantilla describe un conjunto de objetos, describe características tales como parámetros de estado, operaciones y comportamiento. Típicamente, una *instanciación* (creación de una instancia) de un objeto entraña el establecimiento del estado inicial, por ejemplo, un objeto de memoria tampón podría crearse vacío.

El concepto de compatibilidad en comportamiento se aplica también a plantillas de objetos en el sentido de que hay una compatibilidad en comportamiento entre dos plantillas de objetos si los objetos instanciados a partir de esas plantillas son compatibles en comportamiento.

Un *tipo de plantilla* es un predicado definido en una plantilla. Un tipo de plantilla es satisfecho por todas las instancias a partir de la plantilla, y, en general, puede ser satisfecho por otras cosas, cuando éstas cumplen los mismos requisitos que las instancias. Por ejemplo, un tipo de plantilla puede definirse de modo que los objetos instanciados a partir de plantillas diferentes, pero que satisfagan ese tipo de plantilla, muestren compatibilidad en comportamiento.

Cada plantilla da lugar a una *clase de plantilla*: el conjunto de *instancias* del tipo de plantilla. Las clases de plantilla pueden organizarse en jerarquías de subclases, de acuerdo con las relaciones de tipo/subtipo entre los tipos de plantillas.

### 7.2.5 Roles (o papeles)

Un *rol* (o *papel*) identifica, en una plantilla para un objeto compuesto, un comportamiento que ha de asociarse con uno de los objetos componentes.

Un rol puede corresponder a un subconjunto del comportamiento total de un objeto componente. Cuando un objeto se visualiza en términos de un rol, sólo ofrece interés un subconjunto denominado de sus acciones, y se hace abstracción de otras acciones, que posiblemente correspondan a otros roles. Un objeto componente puede tener varios roles en un momento dado, los que dependerán de sus interacciones, y puede desempeñar diferentes roles en tiempos diferentes. Estos roles pueden estar asociados con interfaces.

Por ejemplo, un objeto puede tener su rol normal funcional, o de misión (es decir, el que corresponde a su finalidad) y, para fines de gestión, tener un rol de gestión (es decir, el comportamiento necesario para supervisar y controlar el comportamiento del rol de misión). Cada rol tiene su propia interfaz, en la que el rol de misión está asociado con una interfaz de misión y el rol de gestión está asociado con una interfaz de gestión.

### 7.2.6 Clases de base y clases derivadas

Los conceptos de *clase de base* y *clase derivada* se basan en una noción general de la modificación de plantillas conocida como *modificación incremental*. La modificación incremental es la derivación de una nueva plantilla mediante la modificación de una plantilla existente. La nueva plantilla se llama la plantilla derivada, y la plantilla original se llama la plantilla de base. Las instancias de la plantilla original y de la plantilla derivada se llaman la clase de base y la clase derivada, respectivamente.

En general, una modificación incremental que permite sustitución puede producir una jerarquía diferente de la de clase/subclase.

Por ejemplo, considérese una clase C1 (de plantilla) de automóviles rojos definidos por una plantilla Temp 1 que contiene la siguiente línea:

COLOUR=RED.

Supóngase que para obtener la clase de plantilla C2, esta línea se sustituye por:

COLOUR=BLUE.

La clase de plantilla C2 es una clase derivada de C1 y no una subclase de C1.

En algunos casos, la implementación de una clase derivada puede basarse en la implementación de la clase de base. Este concepto se conoce por herencia de implementación, y permite la compartición de código en programas ejecutables. Sin embargo, esto puede causar problema en un entorno distribuido, pues los cambios introducidos en el código de la clase de base tienen que ser propagados para actualizar todas las implementaciones de las clases derivadas, y, por lo tanto, los sistemas ODP no están obligados a soportar la herencia de implementación.

## 7.3 Conceptos de estructuración

### 7.3.1 Grupos y dominios

Un *grupo* es un conjunto de objetos agrupados por razones estructurales o porque los comportamientos de los objetos tienen características comunes (por ejemplo, pueden sustituirse unos a otros en un grupo de replicación, participan en la misma interacción en un grupo de comunicación). El concepto de grupo es genérico y permite la especificación de diferentes clases de grupos que pueden utilizarse en sistemas distribuidos para muchas finalidades diferentes, tales como tolerancia de *avería*, disponibilidad y soporte de aplicación (por ejemplo en aplicaciones de conferencias).

Un *dominio* es una forma particular de grupo en la cual un aspecto particular del comportamiento de objetos en el grupo es controlado por una misma autoridad. Por ejemplo, en un *dominio de seguridad*, las políticas de seguridad aplicables al comportamiento de los objetos en el dominio son establecidas por la misma *autoridad de seguridad*. El concepto de dominio permite introducir las nociones de autonomía, autoridad y control en sistemas distribuidos. El concepto de dominio responde a muchas necesidades diferentes, ya que los sistemas distribuidos utilizan muchas clases de dominios (dominios de seguridad, dominios de gestión, *dominios de denominación*).

### 7.3.2 Denominación

Se necesita la denominación para distinguir y ganar acceso a componentes de un sistema distribuido, por lo que es un elemento fundamental en la construcción de sistemas distribuidos.

Una denominación dependiente del contexto y una gestión de *nombres* son necesarias para tratar la heterogeneidad, la autonomía y la *federación*. Con esto se obtiene flexibilidad y evolución al permitirse el desarrollo independiente de sistemas de nombres y combinaciones arbitrarias de sistemas de nombres, incluyendo nombres existentes, independientes. Permite también la heterogeneidad en sistemas de nombres, en varios niveles, por ejemplo formas de nombres, asignación de nombres, políticas de denominación y estrategias para la *resolución de nombre* pueden ser diferentes en sistemas diferentes.

Se utilizan nombres para hacer referencia a entidades en un contexto dado. Puede haber circunstancias en que un nombre haga referencia a más de una entidad. Un nombre que hace referencia inequívocamente a una entidad se llama un *identificador*.

Los conceptos de denominación en la Rec. UIT-T X.902 | ISO/CEI 10746-2 no especifican un marco de denominación completo para ODP. Tal marco es un subconjunto para una normalización aparte.

### 7.3.3 Contrato

Un *contrato* es un acuerdo que rige la cooperación entre varios objetos, y corporifica las ideas de *obligación*, *permiso*, *prohibición* y *expectación* asociadas con objetos cooperantes. Por tanto, es un concepto general para caracterizar y regular la cooperación de objetos.

Cuando unos cooperan (interactúan), existe entre ellos algún contrato. En aquellos casos en que el contrato puede ser convenido en algún momento y terminado posteriormente, se está en presencia de una especificación dinámica de la configuración de los objetos. Aunque la cooperación potencial entre los objetos es constante, las reglas proporcionadas por el contrato constriñen la cooperación potencial a algún comportamiento actual, transitorio.

Sin embargo, los contratos a menudo pueden derivarse de las reglas del *lenguaje de punto de vista* y no es necesario enunciarlos explícitamente. Por ejemplo, en el caso del lenguaje computacional, se obliga a un cliente a no invocar una operación que no esté definida en una interfaz de servidor. Los únicos contratos que hay que expresar explícitamente son los que especifican constricciones adicionales.

A título de ejemplo, un contrato puede especificar:

- los roles de objetos y las obligaciones aplicables a los roles, es decir, el comportamiento cooperativo esperado;
- los aspectos de *calidad de servicio* de la cooperación de objetos (cuestiones de seguridad de funcionamiento, corrección, etc.);
- el tipo de comportamiento que *invalida* el contrato.

Un *contrato de entorno* es un tipo particular de contrato que se aplica entre un objeto y su entorno. Este contrato describe los requisitos impuestos por el objeto a su entorno y viceversa. En particular, contiene las constricciones de calidad de servicio (QOS, *quality of service*).

### 7.3.4 Enlace y vinculación

El *comportamiento de vinculación* establece un *contexto contractual* (una vinculación) entre interfaces y permite la cooperación de los objetos. Una vinculación puede existir en varios niveles de abstracción.

Un *enlace (liaison)* es una relación que existe entre los objetos que cooperan en el marco de una vinculación. Cuando se produce un enlace, un objeto sabe que los demás objetos en el enlace se rigen por el contrato. Un objeto puede intervenir en varios enlaces simultáneos: para cada uno de estos enlaces hay un contrato correspondiente.

NOTA – Los términos siguientes, utilizados en el ejemplo del párrafo que sigue pertenecen a la terminología de la interconexión de sistemas abiertos: vinculación, respuesta de vinculación, desvinculación y contexto de aplicación.

Un ejemplo de enlace lo proporciona el comportamiento de vinculación de asociación OSI. El *comportamiento de establecimiento* lo proporciona una operación de vinculación que incluye el envío de parámetros del contrato por el objeto de aplicación iniciador al objeto de aplicación respondedor. El *objeto respondedor* responde con una operación de respuesta de desvinculación (cuyos parámetros de contrato pueden ser diferentes), por lo que se establece el enlace. El contexto contractual para el enlace viene dado por el contexto de aplicación convenido mediante la vinculación en el intercambio de respuesta de vinculación. Cuando una cualquiera de las partes desea terminar el enlace, inicia el *comportamiento de terminación* invocando la operación de desvinculación.

## 8 Arquitectura

La Rec. UIT-T X.903 | ISO/CEI 10746-3 formula los enunciados prescriptivos que deben respetarse para que un sistema sea caracterizado como un sistema ODP. Utilizando los conceptos y la terminología definidos por la Rec. UIT-T X.902 | ISO/CEI 10746-2, estos enunciados definen:

- un marco arquitectural para estructurar la especificación de sistemas ODP en términos de los conceptos de *puntos de vista y especificaciones de punto de vista, y transparencias de distribución*;
- un conjunto de *lenguajes* en cuyos términos pueden expresarse las diferentes especificaciones de punto de vista;
- una infraestructura de sistema que proporciona *transparencias de distribución* para aplicaciones de sistemas.

### 8.1 Marco arquitectural

Los sistemas distribuidos pueden ser muy grandes y complejos, y las numerosas y diferentes consideraciones que pueden influir en su diseño pueden tener por consecuencia un cuerpo sustancial de especificación, al que se le debe dar una estructura si ha de ser gestionado satisfactoriamente. Un buen marco debe permitir trabajar separadamente sobre las diferentes partes del diseño, si son independientes, pero debe identificar claramente los lugares en que los diferentes aspectos del diseño se constriñen unos a otros. Para conseguir esto, en la arquitectura ODP se utilizan dos métodos de estructuración principales: la definición de puntos de vista y la definición de transparencias.

### 8.1.1 Puntos de vista

Un punto de vista es una subdivisión de la especificación de un sistema completo, establecida para reunir las piezas de información particulares relativas a algún tema determinado durante el diseño del sistema. Un sistema ODP es cualquier cosa que ofrezca interés, por lo que, por ejemplo, puede muy bien ser un sistema de procesamiento de información de una organización, o un componente particular (soporte físico o lógico) de ese sistema. Los puntos de vista no son completamente independientes: los elementos esenciales en cada punto de vista se identifican como relacionados con elementos en otros puntos de vista. Sin embargo, los puntos de vista son lo suficientemente independientes para simplificar el razonamiento sobre la especificación completa.

Cada uno de los puntos de vista en el conjunto puede estar relacionado con todos los demás. Los puntos de vista no forman una secuencia fija como un conjunto de capas de protocolo, ni se crean en un orden fijo de acuerdo con alguna metodología de diseño. La arquitectura se expresa en términos del conjunto completo de puntos de vista conexos, sin dejar establecida la manera de construir una especificación completa para un sistema dado cualquiera.

El modelo de referencia de procesamiento distribuido abierto (RM-ODP) define cinco puntos de vista. Estos son:

- a) El *punto de vista de la empresa*: Un punto de vista sobre el sistema y su entorno que destaca la finalidad, alcance y políticas para el sistema.
- b) El *punto de vista de la información*: Un punto de vista sobre el sistema y su entorno que destaca la semántica de la información y el procesamiento de información realizado.
- c) El *punto de vista computacional*: Un punto de vista sobre el sistema y su entorno que permite la distribución mediante una descomposición funcional del sistema en objetos que interactúan en interfaces.
- d) El *punto de vista de la ingeniería*: Un punto de vista sobre el sistema y su entorno que destaca el mecanismo y las funciones requeridas para soportar la interacción distribuida entre objetos en el sistema.
- e) El *punto de vista de la tecnología*: Un punto de vista sobre el sistema y su entorno que destaca las opciones de tecnología elegidas en ese sistema.

Para representar un sistema ODP a partir de un punto de vista determinado es necesario definir un conjunto estructurado de conceptos en cuyos términos puede expresarse esa representación (o especificación). Este conjunto de conceptos proporciona un lenguaje para escribir especificaciones de sistemas a partir de ese punto de vista y tal especificación constituye un modelo de un sistema en términos de los conceptos. Los términos de cada *lenguaje de punto de vista*, y las reglas aplicables a la utilización de esos términos se definen empleando los conceptos de modelado de objeto definidos en la Rec. UIT-T X.902 | ISO/CEI 10746-2. Cada lenguaje tiene un poder de expresión suficiente para especificar una *función ODP*, aplicación o política a partir del punto de vista correspondiente. Cuando las especificaciones de sistema son conformes con estos lenguajes, los sistemas diseñados son sistemas ODP, por lo menos desde un punto de vista arquitectural.

En el RM-ODP, los diversos lenguajes de punto de vista difieren en la fuerza de las constricciones que implica su utilización. Los lenguajes de punto de vista empleados para organizar la distribución y proporcionar soluciones comunes a sus problemas (los puntos de vista computacional y de ingeniería) imponen un número considerable de constricciones que deben ser respetadas y, al hacer esto, garantizan el interfuncionamiento entre componentes, y la portabilidad de éstos. Por otro lado, puesto que el RM-ODP es genérico, las reglas que hay que enunciar en los lenguajes de empresa y de información son pocas, y se limitan a un conjunto de conceptos y directrices básicos sobre el alcance del modelado de la empresa y de la información para sistemas distribuidos.

Se definirán constricciones más extensas para los lenguajes de empresa y de información cuando éstos se utilicen en la especificación de sistemas dentro de campos de aplicación particulares y empresas específicas. Por ejemplo, para cualquier sistema comerciante (*trader system*), las especificaciones de empresa y de información están constreñidas por las disposiciones de la norma del comerciante.

### 8.1.2 Transparencias de distribución

Cuando se diseña un sistema distribuido, surgen varias cuestiones que son resultado directo de la distribución: los componentes del sistema son heterogéneos, pueden fallar independientemente, se encuentran situados en lugares diferentes, que posiblemente varíen, y otras similares. Estas cuestiones pueden o bien resolverse directamente como parte del diseño de la aplicación, o mediante soluciones estándar escogidas basadas en la práctica.

Si se eligen mecanismos estándar, el diseñador de la aplicación trabaja en un mundo que es transparente para esa cuestión particular; se dice que el mecanismo estándar proporciona una *transparencia de distribución*. Los diseñadores de aplicaciones, simplemente, seleccionan las transparencias de distribución que desean emplear y determinan en que punto del diseño deberán aplicarse.

El método de la transparencia de distribución puede conducir directamente a una reutilización del soporte lógico. La selección de transparencias de distribución en la especificación del sistema puede conducir a la incorporación automática de implementaciones bien conocidas de soluciones estándar mediante las herramientas de construcción de sistemas que se están utilizando, tales como compiladores, programas de enlace de módulos para la ejecución (*linkers*) y gestores de configuración. El diseñador establece los requisitos del sistema en forma de un enunciado simplificado del sistema requerido y las propiedades de transparencia de distribución que debe tener.

Las transparencias de distribución definidas en el RM-ODP son:

- a) *transparencia de acceso*, que enmascara las diferencias en la representación de *datos* y mecanismos de *invocación* para permitir el interfuncionamiento de objetos. Esta transparencia de distribución resuelve muchos de los problemas de interfuncionamiento entre sistemas heterogéneos, y generalmente se proporcionará por defecto.
- b) *transparencia de fallo*, que enmascara, con respecto a un objeto, el fallo y la posible *recuperación* de otros objetos (o del mismo objeto) para permitir la tolerancia a las averías. Cuando se proporciona esta transparencia de distribución, el diseñador puede trabajar en un mundo ideal en el que no se produce la correspondiente clase de fallos.
- c) *transparencia de ubicación*, que enmascara la utilización de información sobre *ubicación en el espacio* cuando hay una identificación y vinculación a interfaces. Esta transparencia de distribución proporciona una visión lógica de una denominación, independiente de la ubicación física real.
- d) *transparencia de migración*, que enmascara, con respecto a un objeto, la aptitud de un sistema cambiar la ubicación de ese objeto. La migración se utiliza a menudo para obtener el equilibrio de la carga y reducir la latencia.
- e) *transparencia de reubicación*, que enmascara la reubicación de una interfaz con respecto a otras interfaces ligadas a ella. La reubicación permite que la operación del sistema continúe aunque la migración o la sustitución de algunos objetos produzcan inconsistencias temporales en la visión percibida por sus usuarios.
- f) *transparencia de replicación*, que enmascara la utilización de un grupo de objetos mutuamente compatibles en comportamiento, para soportar una interfaz. La replicación suele utilizarse para mejorar la calidad de funcionamiento y la disponibilidad.
- g) *transparencia de persistencia*, que enmascara, con respecto a un objeto, la *desactivación* y *reactivación* de otros objetos (o del mismo objeto). La desactivación y reactivación se utilizan a menudo para mantener la *persistencia* de un objeto cuando el sistema no pueda proporcionarle continuamente funciones de procesamiento, almacenamiento y comunicación.
- h) *transparencia de transacción*, que enmascara la coordinación de actividades dentro de una configuración de objetos, para asegurar la consistencia.

En toda especificación de sistema, la definición de la transparencia implica un conjunto de requisitos y una transparencia de distribución que lo satisface. El conjunto de requisitos indica dónde se necesita la transparencia de distribución (es decir, las interacciones que son afectadas). Esto puede ser, simplemente, un enunciado aplicable a todo el sistema, o un enunciado más selectivo que indique interfaces específicas y defina, por ejemplo, las interacciones que pueden constituir una transacción, o seleccionar los objetos e interfaces que habrán de ser soportados por replicación. La solución puede ser en forma de un conjunto de reglas para transformar la especificación de la transparencia de distribución requerida en una especificación en la que las interacciones u objetos seleccionados se amplíen para incluir mecanismos que proporcionen dicha transferencia.

## 8.2 Lenguaje de empresa

El lenguaje de empresa introduce conceptos básicos necesarios para representar un sistema ODP en el contexto de la empresa en que funciona. La finalidad de una especificación de empresa es expresar los objetivos y las constricciones impuestas por las políticas en el sistema de que se trate. Para ello, el sistema se representa por uno o más objetos de empresa dentro de una *comunidad* de objetos de empresa que representa la empresa y por los roles en que intervienen estos objetos. Estos roles representan, por ejemplo, los usuarios, dueños y proveedores de información procesada por el sistema. La creación de un punto de vista separado para transportar esta información desacopla la especificación de los objetivos fijados para un sistema, de la manera en que dicho sistema ha de realizarse.

Una de las ideas esenciales en el lenguaje de empresa es la de un contrato, que asocia a los que desempeñan diversos papeles en una comunidad y expresa sus obligaciones mutuas. Un contrato puede expresar las metas y responsabilidades comunes que caracterizan roles en una comunidad, como un negocio y sus clientes o una organización gubernamental y sus clientes, tal como están relacionados de maneras particulares en una misma empresa.

Cuando proceda, una especificación de empresa expresará también aspectos relativos a la propiedad de los recursos y la responsabilidad por el pago de mercancías y servicios para identificar, por ejemplo, constricciones supuestas a los mecanismos de contabilidad y seguridad dentro de la infraestructura que soporta al sistema.

Un género particular de comunidad es una federación, que es la reunión de varios grupos que responden a autoridades diferentes (por lo que pueden representarse como dominios distintos), de modo que, juntos, puedan cooperar para alcanzar algún objetivo. Dado que la evolución de los sistemas distribuidos tendrá repetidamente por consecuencia la fusión de subsistemas existentes, gestionados separadamente, para compartir información o apoyar intereses comerciales, la especificación de la *creación* de federaciones y la expresión de las reglas que deban regirlas constituye una parte importante de la especificación de sistema en el punto de vista de la empresa.

Los dominios reunidos en una federación pueden ser dominios administrativos (cada uno sujeto, por ejemplo, a determinados controles de seguridad y gestión) o dominios de tecnología (cada uno sujeto, por ejemplo, a decisiones comunes sobre las opciones relativas los soportes físico y lógico del sistema). La especificación de una federación implica la especificación de los objetivos para el interfuncionamiento entre diferentes dominios y de las políticas que rigen ese interfuncionamiento.

La federación de dominios administrativos se relaciona con el interfuncionamiento entre los dominios en la misma empresa o en empresas diferentes para proporcionar la compartición, integración o división de recursos y aplicaciones entre (o en) los diferentes sistemas y ubicaciones, en respuesta a las necesidades de los usuarios. La federación de dominios de tecnología se refiere a la integración de diferentes arquitecturas de sistemas y de sistemas con diferentes recursos y diferentes calidades de funcionamiento; proporciona la modularidad que permite un crecimiento incremental sin que esto influya en las aplicaciones existentes. Estas dos clases de federación a menudo coinciden, pues las diferencias en administración pueden conducir a diferencias en la elección de la tecnología.

Entre los dominios administrativos, cada una de las administraciones, o ambas, pueden desear imponer sus propios controles de acceso para fines tales como la seguridad, la contabilidad y la supervisión, además de los controles impuestos por los propios objetos. Las fronteras administrativas son también los puntos en que se producen los cambios de la responsabilidad de la gestión para asuntos tales como la atribución de recursos y las garantías de seguridad de funcionamiento.

Las políticas que rigen la operación de una federación comprenden las políticas que rigen el interfuncionamiento. Por tanto, la especificación de una federación puede estar relacionada con la necesidad de especificar facilidades de *interceptor* en la descripción de ingeniería, y los objetivos y la política para la federación imponen constricciones a la provisión de facilidades de interceptor.

Una especificación de empresa define las políticas que rigen el comportamiento de las comunidades en ella especificadas. Estas políticas determinan las acciones de los objetos de empresa que forman esas comunidades, y se encargan de imponer y cumplir esas obligaciones (por ejemplo, solicitar una entrega, ejecutar una entrega), así como de permitir y prohibir acciones (por ejemplo, autorizar o rechazar el acceso a facilidades del sistema). Las políticas pueden relacionarse con:

- a) La estructuración de la comunidad en términos de roles y la asignación de roles a objetos de empresa. Por ejemplo, las **reglas de comunidad** pueden expresar:
  - asignación de roles y responsabilidades a objetos de empresa dentro de la comunidad;
  - la manera en que los objetos de empresa están relacionados en la estructura de la comunidad (por ejemplo, jerarquía o isocracia).

Una especificación de empresa puede también incluir **reglas mercantiles** que indican:

- la empresa como una entidad mercantil;
  - requisitos de contabilidad;
  - la evolución de los negocios para la consecución de sus objetivos.
- b) Interacciones permitidas entre objetos de empresa que desempeñan roles diferentes (esto es, control de acceso). Por ejemplo, las **reglas de seguridad** pueden definir:
    - las relaciones rol-*actividad*-objeto y sus requisitos de integridad y confidencialidad para actividades y objetos;
    - las reglas para la detección de amenazas a la seguridad;
    - las reglas para la protección contra las amenazas a la seguridad;
    - las reglas para limitar los daños causados por las eventuales infracciones a la seguridad.

- c) La responsabilidad delegada a objetos de empresa. Por ejemplo, se utilizan **reglas de delimitación de la autoridad** para asignar:
- privilegios a objetos de empresa (trust);
  - permiso o prohibición de acciones de objetos de empresa.
- d) La contabilidad relativa a la utilización de recursos. Por ejemplo, las **reglas de utilización de recursos** definen las constricciones que pueden ser impuestas por:
- organismos reguladores;
  - exigencias del mercado;
  - entorno,
- externos, lo que dependerá de que la utilización de los recursos sea:
- pública;
  - privada;
  - por terceros.
- e) La propiedad de los recursos. Por ejemplo, las **reglas de transferencia** pueden indicar el intercambio de la propiedad y/o responsabilidades sobre los recursos entre los objetos de empresa.
- f) Los miembros de las federaciones. Por ejemplo, una especificación de empresa puede incluir **reglas de los dominios** que especifiquen:
- las reglas que determinan la calidad de miembro de un dominio;
  - las reglas de interacción entre dominios del mismo tipo;
  - las reglas de denominación de dominios.

Se espera que habrá diferentes notaciones para expresar especificaciones de empresas para estructuras organizacionales y prácticas de negocios específicas. El RM-ODP requiere que se elabore una especificación apropiada, pero impone pocas constricciones en cuanto a la forma que deben adoptar las organizaciones.

La evaluación de la conformidad con la especificación de empresa de un sistema implica que los requisitos (por ejemplo, un tiempo de respuesta para el cumplimiento de una obligación) expresados en la especificación se relacionen con conjuntos de observaciones del comportamiento del sistema en *puntos de conformidad* identificados en la especificación de ingeniería y en la especificación de tecnología, y la evaluación del grado de consistencia entre los requisitos y las observaciones.

### 8.3 Lenguaje de información

Los componentes individuales de un sistema distribuido tienen que entender de una manera común la información que se comunican cuando interactúan, pues de lo contrario el sistema no se comportará de la manera esperada. Algunos de estos elementos de información son manipulados, de una u otra forma, por muchos de los objetos en el sistema. Para asegurar que la interpretación de estos elementos de información es consistente, el lenguaje de información define conceptos para la especificación del significado de la información almacenada en un sistema ODP y manipulada por dicho sistema, independientemente de la manera en que las funciones de procesamiento de la información propiamente dichas deban ser implementadas.

La información contenida en el sistema ODP sobre entidades en el mundo real, incluidas las relativas al propio sistema ODP, se representa, en una especificación de información, en términos de objetos de información, y sus relaciones y comportamientos. Los elementos de información básicos se representan por objetos de información atómica. La información más compleja se representa como objetos de información compuestos, cada uno de los cuales expresa relaciones relativas a un conjunto de objetos de información constituyentes.

Al igual que en el modelado usual de datos, la especificación de información, comprende un conjunto de esquemas conexos, a saber, los esquemas *invariante, estático y dinámico*.

Un *esquema invariante* expresa relaciones entre objetos de información que siempre tienen que ser verdaderas, para todos los comportamientos válidos del sistema. Así, un esquema invariante para una cuenta bancaria podría especificar que el saldo tiene que ser siempre positivo, o cero, pues el banco no ofrece la posibilidad de sobregiro.

Un *esquema estático* expresa aserciones que tienen que ser verdaderas en un solo instante. Los esquemas estáticos suelen utilizarse para especificar el estado inicial de un objeto de información. Por ejemplo, el estado inicial de un objeto de cuenta bancaria consiste en un saldo de cuenta de \$0 y la cantidad extraída ese día es también \$0. Podría utilizarse otro esquema estático para describir cómo la cantidad extraída ese día es \$0 a las doce de la noche, cada día; obsérvese que este esquema estático no establece restricción alguna en cuanto al saldo de la cuenta en ese instante.

Un *esquema dinámico* especifica la forma en que la información puede evolucionar en el transcurso de la operación del sistema. Por ejemplo, una cuenta bancaria requeriría un esquema dinámico para el depósito de dinero, la extracción de dinero, el pago de intereses, y gastos bancarios debitados a la cuenta. Un esquema dinámico podría ser aplicable solamente en ciertas circunstancias (que podrían especificarse por la utilización de un esquema estático). Por ejemplo, el esquema dinámico para extraer \$N podría especificar que el saldo de la cuenta se redujera en \$N siempre que la cantidad total extraída ese día no fuese superior a \$500. Ningún esquema dinámico puede especificar un estado resultante que viole la restricción invariante; es decir, de una cuenta sólo puede extraerse dinero.

Además de describir los cambios de estado, los esquemas dinámicos pueden también crear y suprimir objetos componentes. Esto permite modelar una especificación de información completa de un sistema ODP como un objeto de información (compuesto) único.

Estos esquemas pueden aplicarse a la totalidad del sistema o a determinados dominios, dentro del mismo. En particular, en los sistemas grandes que evolucionan rápidamente, la reconciliación y federación de dominios de información distintos será una de las tareas principales que habrán de emprenderse para gestionar la información.

Los esquemas para los objetos de información compuestos no tienen que hacer referencia a todos los componentes del objeto de información. Los esquemas para objetos de información compuestos pueden estar formados por esquemas para sus objetos componentes a condición de que dicha composición tenga sentido. La encapsulación de objetos de información se relaciona con el nivel de abstracción de la descripción en cuestión. Así, en un nivel apropiado de abstracción, los esquemas para objetos de información compuestos pueden hacer referencia a los elementos internos de sus objetos componentes, aunque, en un nivel de abstracción más alto, esto puede no ser posible. Esto permite la especificación de frases substantivas complejas tales como «los números telefónicos de los clientes con cuentas en la que se hayan extraído más de \$400 hoy».

Algunos elementos visibles desde el punto de vista de la empresa serán visibles desde el punto de vista de la información, y viceversa. Por ejemplo, una actividad percibida desde el punto de vista de la empresa puede aparecer en el punto de vista de información como la especificación de algún procesamiento que provoca una transición de estado de una entidad de información.

Notaciones diferentes para especificaciones de información modelan las propiedades de la información de maneras diferentes. Puede hacerse hincapié en la clasificación y reclasificación de tipos de información o en los estados y el comportamiento de objetos de información. En algunos lenguajes de especificación, los objetos de información atómica se representan como valores. El método que ha de seguirse dependerá de la técnica de modelado y de la notación que se utilicen.

La evaluación de la conformidad con la especificación de información de un sistema implica el establecimiento de una relación entre los requisitos expresados en la especificación (por ejemplo, en un esquema invariante) y conjuntos de observaciones del comportamiento del sistema en puntos de conformidad identificados en las especificaciones de ingeniería y de tecnología, y la evaluación del grado de consistencia entre los requisitos y las observaciones.

## 8.4 Lenguaje computacional

El punto de vista computacional está directamente relacionado con la distribución del procesamiento, pero no con los mecanismos de interacción que hacen posible que se produzca la distribución. La especificación computacional descompone el sistema en objetos que realizan funciones individuales e interactúan en interfaces bien definidas. Proporciona así la base para decisiones sobre la manera de distribuir los trabajos que habrán de realizarse, porque las interfaces pueden estar situadas independientemente si se supone que los mecanismos de comunicación pueden definirse en la especificación de ingeniería para soportar el comportamiento en esas interfaces.

La parte principal del lenguaje computacional es el modelo de objeto, que define:

- la forma de interfaz que puede tener un objeto;
- la manera en que las interfaces pueden ser vinculadas y las formas de interacción que puede tener lugar en ellas;
- las acciones que puede ejecutar un objeto, en particular la creación de nuevos objetos o interfaces, y el establecimiento de vinculaciones.

El modelo de objeto computacional proporciona la base para asegurar que los lenguajes de especificación, los lenguajes de programación y los mecanismos de comunicación funcionen, todos ellos, de una manera coherente, permitiendo así el interfuncionamiento abierto y la portabilidad de los componentes.

El lenguaje computacional permite al especificador imponer constricciones a la distribución de una aplicación (en términos de contratos de entorno asociados con interfaces individuales y vinculaciones de interfaz) sin especificar el grado de distribución real. Con esto se asegura que las aplicaciones no contengan supuestos no enunciados que afecten a la distribución de sus componentes. Como consecuencia de esto, la configuración y el grado de distribución del soporte físico con el que se ejecutan las aplicaciones ODP pueden ser fácilmente modificados, siempre que se cumplan las condiciones enunciadas para el entorno, sin que ello tenga mayores consecuencias en el soporte lógico de la aplicación.

El lenguaje computacional no excluye la utilización, en un entorno distribuido, de soportes lógicos diseñados para sistemas centralizados. Permite la encapsulación de aplicaciones existentes como componentes (no distribuidos) de una aplicación mayor, distribuida. Esto permite aplicar un método evolutivo a la provisión de la distribución, con lo que se protegen las inversiones hechas en el soporte lógico existente.

Las interacciones entre objetos computacionales son esencialmente asíncronas y pueden ser de tres formas:

- *operaciones*, que son similares a procedimientos y son invocadas en interfaces designadas;
- *flujos*, que son abstracciones de secuencias continuas de datos entre interfaces;
- *señales*, que son interacciones atómicas elementales.

Las operaciones reflejan el paradigma cliente/servidor. Una operación es una interacción entre un *objeto de cliente* y un *objeto de servidor* por la que se pide (una invocación) que el servidor realice alguna función. Hay dos tipos de operaciones:

- una *interrogación*, en la que el servidor retorna una respuesta (una *terminación*) a la petición del cliente;
- un *anuncio*, en el que no hay respuesta a la petición del cliente.

La noción de terminación generaliza los resultados y excepciones que se encuentran en muchos lenguajes de programación basados en objetos y no basados en objetos.

La realización de operaciones se extiende en el espacio y en el tiempo. En consecuencia, cuando una operación fracasa, el fallo no tiene necesariamente que producirse para todos los participantes y puede ser observado por los participantes en momentos diferentes. La aptitud del cliente para observar fallos y realizar acciones cuando se producen fallos es diferente para las interrogaciones y los anuncios.

En el caso de una interrogación, la toma de contacto bidireccional asegura que el cliente recibe una confirmación de que la función solicitada ha sido realizada, y que, si un *hilo (thread)* de actividad de un cliente invoca una *cadena* de interrogaciones, el servidor responderá a las peticiones en el mismo orden en que fueron emitidas por el cliente.

En el caso de un anuncio, la garantía de la ejecución de las peticiones, y el orden en que éstas se ejecutan, vienen determinadas por los contratos de entorno aplicables a las operaciones.

Se pueden utilizar flujos para modelar, por ejemplo, el flujo de información de audio o vídeo en una aplicación multimédios, o en servicios de telecomunicación basados en la voz, o el flujo continuo de lecturas periódicas de un detector en una aplicación de control de proceso. Un flujo se caracteriza por su nombre y su tipo, que especifica la naturaleza y el formato de los datos intercambiados. La semántica exacta de los flujos se deja sin definir en un modelo computacional. De hecho, puede haber muchas semánticas diferentes para flujos, las que dependerán del dominio de aplicación.

Las señales constituyen el nivel más bajo de descripción de interacciones entre objetos computacionales. Una señal es una acción atómica compartida, por pares, y que produce una comunicación unidireccional de un objeto computacional *iniciador* a un objeto computacional *respondedor* (en este contexto "respondedor" significa "que acepta la comunicación"). Esto significa que:

- la señal se produce en un instante definido y, por tanto, es un punto de referencia para fines de medición (por ejemplo, en observaciones de la calidad de servicio);
- un fallo es idéntico para todos los participantes, que podrán percibirlo.

En muchos casos, una señal corresponderá, en términos de la implementación, a un evento observable en alguna ubicación física; sin embargo, la definición del concepto no excluye la implementación de señales mediante mecanismos de transacción que dan las necesarias garantías de comportamiento.

Una operación o un flujo puede explicarse en términos de una combinación de varias señales. Una interrogación, por ejemplo, puede entenderse como una secuencia de señales: emisión de la invocación (por el objeto de cliente), recepción de la invocación (por el objeto de servidor), emisión de la terminación (por el objeto de servidor), recepción de la terminación (por el objeto de cliente). En cambio, como el modelo computacional no da la semántica exacta de los flujos, su correspondencia con señales no está definida. El modelado de operaciones o flujos en términos de señales se hace necesario para definir características de calidad de servicio de extremo a extremo, y la operación de la vinculación multipartita o las vinculaciones entre diferentes clases de interfaz (por ejemplo, vinculación de un tren a una *interfaz de operación*).

#### 8.4.1 Interfaces computacionales

Una interfaz computacional se caracteriza por una firma, un comportamiento, y un contrato de entorno.

La firma depende del tipo de interfaz, que puede ser de operación, de tren o de señal:

- Una *interfaz de operación* tiene una firma que define el conjunto de operaciones soportadas en la interfaz y determina si la interfaz desempeña el rol de cliente o de servidor para ese conjunto de operaciones.
- Una *interfaz de tren* tiene una firma que define el conjunto de flujos soportados en la interfaz y determina, para cada flujo, si la interfaz desempeña el rol de productor o de consumidor.
- Una *interfaz de señal* tiene una firma que define el conjunto de señales soportadas en la interfaz y determina, para cada señal, si la interfaz desempeña el rol de iniciador o de respondedor.

El comportamiento se describe por las secuencias autorizadas de acciones del objeto computacional que están asociadas con la interfaz. El comportamiento puede incluir acciones internas del objeto y será constreñido por el entorno del objeto, en particular por las interacciones con otras interfaces.

Es importante señalar que cada especificación de interfaz contiene también un contrato de entorno que especifica un conjunto de constricciones de calidad de servicio impuestas a un objeto computacional y su entorno. Si el entorno (otros objetos computacionales e infraestructura de soporte) proporciona el nivel requerido de calidad de servicio, está entonces garantizado (por diseño) que el propio objeto proporciona un cierto nivel de calidad de servicio. La calidad de servicio especificada para una interfaz expresa los requisitos impuestos a su entorno, y la calidad de servicio mostrada por el objeto en un entorno que satisface los requisitos. El lenguaje computacional no prescribe la notación particular para especificar la calidad de servicio.

#### 8.4.2 Modelo de vinculación

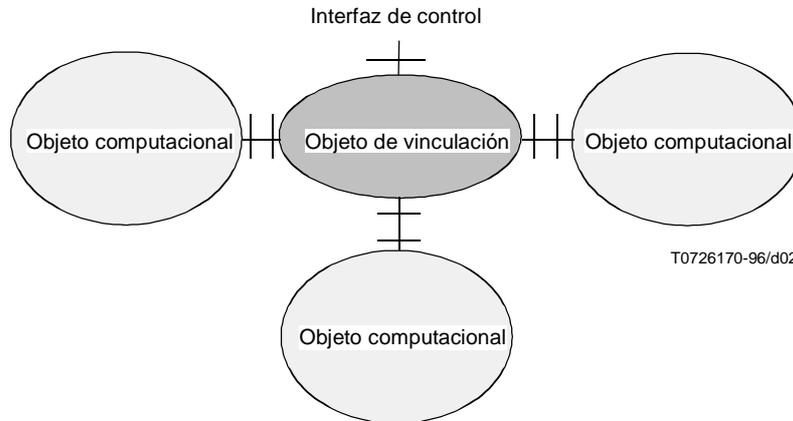
Las interacciones entre interfaces computacionales dadas solo son posibles si se ha establecido entre ellas alguna vinculación (es decir, algún trayecto de comunicación). El lenguaje computacional especifica acciones de *vinculación explícita* para las interfaces de operación y las interfaces de tren. En el caso de interfaces de operación, especifica también que la vinculación puede ser *implícita*, con el fin de permitir la utilización de notaciones que no prevén la expresión de acciones de vinculación. La *vinculación implícita* puede también producirse para interfaces de operación, pues en otros casos no es evidente quién tiene la iniciativa de la vinculación con relación a interacciones subsiguientes.

Cuando la vinculación es implícita, una invocación por el objeto de cliente se traduce por una vinculación de una interfaz de cliente apropiada para la interfaz de servidor y la aparición de la operación (interrogación o anuncio). El lenguaje computacional deja sin definir si la interfaz de cliente se suprime o no al final del proceso. Debe señalarse que la vinculación implícita no prevé una referencia a un contrato de entorno para la vinculación.

Cuando la vinculación es explícita, está definida en términos de dos clases de acciones: *acciones de vinculación primitiva* y *acciones de vinculación compuesta*. Estas acciones de vinculación sólo son aplicables en el contexto de la vinculación explícita.

Una acción de vinculación primitiva permite la vinculación de dos interfaces de dos objetos computacionales iguales o diferentes. Las interfaces deben ser del mismo tipo, pero pueden ser interfaces de operación, de tren o de señal. Una acción de vinculación primitiva es ejecutada por uno de los objetos que intervienen y tiene por efecto establecer, en cada interfaz, la información necesaria para que se produzca una interacción, es decir, la identidad de la otra interfaz que interviene. No se observa ningún requisito que exija la definición de una acción de *desvinculación*, pero al suprimirse cualquiera de las dos interfaces, es evidente que se suprime también la vinculación. Una acción de vinculación primitiva requiere que las interfaces que intervienen sean del mismo tipo y tengan un tipo de firma y roles complementarios (por ejemplo, que una sea de cliente y la otra de servidor).

Una acción de vinculación compuesta permite la vinculación de dos o más interfaces del mismo tipo o de tipos diferentes por medio de un *objeto de vinculación* (véase la figura 2).



**Figura 2 – Vinculación compuesta**

La acción la puede ejecutar uno de los objetos computacionales que intervienen en la vinculación, o un objeto computacional que no interviene en la vinculación. Tiene por efecto la instanciación de un objeto computacional para soportar la vinculación (el objeto de vinculación). El objeto de vinculación instancia un conjunto apropiado de interfaces y utiliza acciones de vinculación primitiva para vincularlas a las interfaces que habrán de ser vinculadas. Instancia también un conjunto de interfaces de control a través de las cuales podrán ser controladas sus operaciones y retorna, al objeto computacional iniciador, los identificadores (de interfaz) para las interfaces.

Los comportamientos de los objetos de vinculación reflejan la semántica de comunicación que soportan y el modelo computacional no limita los tipos de objeto de vinculación, con lo que refleja el hecho de que existe una multiplicidad de posibles estructuras de comunicación entre objetos. Sin embargo, pueden normalizarse clases útiles de objeto de vinculación, que dependerán de las clases de aplicación. En particular, los objetos de vinculación pueden especificar la operación de vinculaciones multidireccionales y de vinculaciones complejas (por ejemplo, entre interfaces de operación y de tren de tipos diferentes, y entre interfaces de operación e interfaces de tren).

Al igual que en el caso de cualquier otro objeto, los objetos de vinculación pueden ser calificados por aserciones relativas a la calidad de servicio que imponen ulteriores constricciones a su comportamiento correcto (por ejemplo, para confinar el retardo de comunicación de extremo a extremo o la fluctuación del retardo de extremo a extremo en una interfaz de receptor). Cuando se hacen estas aserciones de calidad de servicio, las interfaces que intervienen en las vinculaciones primitivas de un objeto de vinculación deben ser interfaces de señal, ya que la naturaleza atómica de las señales hace posible la especificación de puntos en el espacio y en el tiempo en los cuales pueden hacerse observaciones de la calidad de servicio.

Las interfaces de control para un objeto de vinculación permiten la *supresión* de la vinculación y pueden también permitir el control de su funcionamiento y de la calidad de servicio que ofrece. Son ejemplos de las facilidades que podrían proporcionarse:

- a) Control de *notificación* de *errores* que trastornan al objeto de vinculación: Esto permitiría la especificación de una interfaz en la cual el objeto invoca una operación de notificación si se produce un trastorno de la vinculación.
- b) Control de una vinculación dinámica de multidifusión, que permite la adición de nuevos consumidores y la supresión de consumidores existentes.
- c) Invocación de grupo, que hace que invocaciones atómicas de multidifusión estén disponibles en el lenguaje computacional y permite añadir miembros al grupo o separar miembros del grupo.

- d) Control de la calidad de servicio asociada con la vinculación, lo que permite la modificación de características específicas de calidad de servicio: Esta forma de control sería particularmente útil para vinculaciones de tren en aplicaciones multimedia.
- e) Notificación de eventos de interés para la aplicación; por ejemplo, podría señalarse un evento al comienzo o al final de un periodo de silencio en un flujo de audio.

En el caso de vinculación de interfaces de tren, la vinculación puede hacer abstracción de reglas de composición de tren específicas de la aplicación. En el caso más simple, la vinculación representará un flujo único de una interfaz de productor a una interfaz de consumidor (por ejemplo, de un sistema de archivo de audio a un altavoz). Sin embargo, las reglas de composición pueden ser mucho más complejas:

- a) Se puede crear y gestionar un trayecto dúplex como una vinculación única; los flujos resultantes enlazan los aspectos de productor de la interfaz en cada objeto computacional con los aspectos de consumidor de la interfaz en el otro.
- b) Varias interfaces dúplex pueden ser enlazadas por un objeto de vinculación que encapsula las reglas de un sistema de conferencias para permitir que el flujo de un *productor* seleccionado (el locutor actual) se transmita a todos los consumidores. Se puede ejercer diversos grados de control de aplicación a través de una interfaz de control de vinculación para proporcionar un control de flujo explícito.
- c) Los flujos de varios productores pueden combinarse para proporcionar un flujo compuesto a un solo consumidor. Por ejemplo, un flujo de vídeo de una fuente y un flujo de audio de otra fuente podrían combinarse para formar un solo flujo de televisión como una imagen y un comentario asociado. En este caso, la interfaz de control podría permitir la manipulación de mecanismos de sincronización de flujo de ingeniería como parte del suministro de una sincronización de labios.

Una interfaz puede estar múltiplemente ligada. En el caso de vinculación implícita, la vinculación múltiple de una interfaz requiere que la interfaz de servidor identifique cada una de las vinculaciones. En el caso de vinculación explícita, la vinculación múltiple requiere que cada objeto de vinculación que intervenga identifique cada una de las vinculaciones.

### 8.4.3 Tipificación y subtipificación para interfaces computacionales

Las interfaces en el lenguaje computacional están sumamente tipificadas, para maximizar una temprana verificación de la consistencia de los programas distribuidos conforme al lenguaje computacional ODP. Los tipos de interfaz están asociados por una relación de subtipificación que define las condiciones mínimas que habrán de imponerse para proporcionar una interacción significativa de los objetos.

El tipo de firma de una interfaz define la forma y la clase de interacciones disponibles en la interfaz, y la subtipificación de la firma especifica los requisitos mínimos para que una interfaz pueda substituir a otra. Las reglas se basan en la semántica de interacción de las interfaces computacionales y son suficientes para asegurar que una interfaz que substituya a otra pueda interpretar correctamente la estructura de cualquier interacción que se produzca. Desde luego, también es necesario que las interfaces concuerden en términos de la semántica de los datos transferidos, pero no pueden definirse reglas generales para llevar a cabo dicha concordancia.

Una *firma de interfaz de señal* define, para cada señal en la interfaz, su nombre, los parámetros y determina si la interfaz en cuestión es la *iniciadora* o la *respondedora*. Una *firma de interfaz de operación* define, para cada clase de operación en la interfaz, el nombre de la operación, el número y los tipos de sus argumentos, así como, en el caso de interrogaciones, el conjunto de resultados posibles de la operación (terminaciones). Para cada terminación, se define el nombre de la terminación, así como el número y los tipos de sus argumentos.

En el anexo A de la Rec. UIT-T X.903 | ISO/CEI 10746-3 se definen reglas de subtipificación para las firmas de interfaces de señal y de operación. Obsérvese que la semántica de interacción para otros tipos de interfaces, además de las de operación y de tren, pueden expresarse en términos de señales, ya que las señales proporcionan los bloques constructivos a partir de los cuales se modelan los tipos de interacciones de alto nivel.

Una interfaz de tren se tipifica en términos de un conjunto de flujos componentes, cada uno de los cuales tiene un tipo básico que puede ser soportado por el mecanismo subyacente disponible. Son ejemplos de tipos básicos los flujos de vídeo o audio individuales. Cada flujo componente tiene un sentido único de transmisión, es decir, o bien entra, o bien sale de la vinculación. Los flujos componentes son organizados por la descripción de tipo en una firma de tren (de la misma manera en que se organizan los argumentos en una firma, en una operación de interfaz). Una interfaz de tren puede comprender varios flujos conexos, en el mismo sentido o en sentidos opuestos. En el caso de trenes, sin embargo, el propio tipo de interfaz puede utilizarse para representar un flujo más complejo, por lo que, en los tipos construidos en una serie de etapas, los flujos puedan formar una jerarquía de múltiples niveles.

Son ejemplos de tipos de interfaces de trenes:

- a) un flujo de audio único desde una fuente de audio;
- b) un flujo de audio único hacia un sumidero de audio;
- c) un tipo de conversación dúplex en el que hay un flujo entrante y un flujo saliente, para representar la percepción, por el usuario, de los aspectos de audio de un servicio de telefonía;
- d) una señal de televisión compuesta, constituida por un componente de audio y un componente de vídeo;
- e) un tipo orientado a la aplicación, más complejo, en el cual varios flujos de audio y de vídeo se combinan para representar flujos en un sistema de realidad virtual.

El hecho de que cada uno de los flujos en una interfaz de tren tenga un sentido de transmisión implica que los tipos de interfaz, en general, existirán en pares que estarán relacionados por la reversión de todos los flujos que intervienen. Sin embargo, si el tipo de interfaz tiene un conjunto equivalente de flujos entrantes y salientes, los mismos dos tipos conexos serán equivalentes. Una notación de definición de interfaz de tren puede permitir formas abreviadas para indicar esto, o para la definición de las dos formas conexas, simultáneamente.

Todo sistema de tipo de interfaz de corriente tendrá asociado un conjunto de reglas de subtipificación. Estas reglas difieren de las reglas de subtipificación para las interfaces de operación, pues éstas se construyen para permitir la comunicación entre objetos computacionales cuyas interfaces de trenes ofrecen diferentes capacidades. Por ejemplo, una interfaz de audio podría considerarse un subtipo de una interfaz compuesta de audio y vídeo, a fin de permitir que un usuario de telefonía distante comunique con un usuario de un sistema de videofonía. La forma óptima de la subtipificación dependerá de la aplicación, por lo que la selección de una variante adecuada de la subtipificación forma parte del proceso de diseño.

En general, las reglas de subtipificación de trenes pueden subdividirse en dos pasos:

- a) identificación de las correspondencias entre flujos primitivos en los dos tipos, y determinar si las correspondencias halladas son o no suficientes para que exista la relación de subtipificación;
- b) comparación de los tipos de cada uno de los flujos primitivos, incluida la comparación de los aspectos de calidad de servicio, para determinar si existe una relación de subtipo.

No es posible definir completamente reglas generales de subtipificación para interfaces de trenes, ya que dichas reglas dependen de los detalles de las interacciones, de los cuales se hace abstracción en las definiciones de los trenes que intervienen.

#### **8.4.4 Portabilidad**

El lenguaje computacional define las acciones que un objeto puede ejecutar y enumera los posibles modos de fallo de dichas acciones. Así, el lenguaje computacional define un modelo de programación basado en objetos para una máquina virtual genérica que se realiza por reglas de ingeniería y de tecnología.

Pueden definirse diferentes conjuntos de reglas de portabilidad, cada una de las cuales especifica un subconjunto particular de acciones definidas por el modelo de programación computacional. Un conjunto de reglas de portabilidad identifica los requisitos que debe cumplir una notación computacional para soportar la portabilidad de objetos entre diferentes entornos que proporcionan implementaciones de esas reglas. El propio RM-ODP define un entorno de portabilidad básica y un entorno de portabilidad completa, que dependen de los conjuntos de acciones soportados.

## 8.5 Lenguaje de ingeniería

El lenguaje de ingeniería enfoca la manera de lograr la interacción de objetos y los recursos necesarios para ello. Define conceptos para describir la infraestructura requerida para soportar interacciones selectivas transparentes a la distribución entre objetos, y reglas para estructurar *canales* de comunicación entre objetos y para estructurar sistemas para fines de gestión de recursos.

Por tanto, el punto de vista computacional determina cuándo y por qué interactúan los objetos, mientras que el punto de vista de la ingeniería determina cómo interactúan. En el lenguaje de ingeniería, la tarea principal es el soporte de interacciones entre objetos computacionales. En consecuencia, hay enlaces muy directos entre las descripciones de punto de vista: los objetos computacionales son visibles en el punto de vista de la ingeniería como *objetos de ingeniería básicos* y las vinculaciones computacionales, sean implícitas o explícitas, son visibles como canales o como vinculaciones locales.

Los conceptos y reglas son suficientes para permitir la especificación de interfaces internas dentro de la infraestructura, lo que permite la definición de puntos de conformidad distintos para transparencias diferentes, y la posibilidad de normalización de una infraestructura genérica en la cual puedan situarse módulos de transparencia normalizados.

### 8.5.1 Conglomerados, cápsulas y nodos

El lenguaje de ingeniería trata los objetos de ingeniería básicos y diversos objetos de ingeniería que los soportan. Relaciona estos objetos con los recursos de sistema disponibles, para lo cual identifica una serie de agrupaciones anidadas (es decir, que unos elementos están contenidos en otros).

En el nivel más exterior, los objetos de ingeniería están físicamente ubicados y asociados con recursos de procesamiento, agrupándolos en *nodos* que pueden ser imaginados como representativos de sistemas de computación manejados independientemente. Un nodo puede ser cualquier cosa que tenga una visión intensamente integrada de los recursos, mientras el diseñador del sistema pueda considerarlo como un todo. Por tanto, un sistema de procesamiento en paralelo estrechamente acoplado puede considerarse un nodo, en tanto en cuanto tenga una sola política de calendarización (*scheduling*) y asignación (un sistema operativo único).

El nodo es controlado por un *núcleo* que es responsable de la iniciación, la creación de grupos de objetos de ingeniería, la puesta a disposición de los medios de comunicación, y la provisión de servicios básicos como la temporización y la creación de identificadores únicos.

Dentro de un nodo puede haber un número de *cápsulas*. Una cápsula tiene su propio almacenamiento y comparte los recursos de procesamiento del nodo. Puede considerarse como si fuese un proceso protegido tradicional, con su propio espacio de direcciones. Una cápsula es, por tanto, la unidad de protección y generalmente es la unidad más pequeña de fallo independiente soportado por el sistema operativo. Hay un objeto de ingeniería especial, denominado el *gestor de cápsula*, asociado con cada cápsula y, para fines descriptivos, una cápsula es controlada por interacciones con este gestor.

Una cápsula contendrá usualmente muchos objetos de ingeniería; la agrupación de objetos en cápsulas se hace para reducir el costo de la interacción de objetos. Esto se debe a que la comunicación entre procesos tradicionales es lenta y costosa, por las comprobaciones que hay que efectuar; sin embargo, se puede confiar a las herramientas de compilación que construyen las cápsulas la *validación* y la estructuración de interacciones entre objetos de ingeniería estrechamente relacionados, en un grado suficiente que les permita compartir recursos. Los recursos dentro de una cápsula serán controlados por cierto tipo de sistema para la fase de ejecución, específico del lenguaje.

La agrupación más pequeña de objetos de ingeniería es en un conjunto de conglomerados dentro de una cápsula. Los objetos que forman el conglomerado se agrupan para reducir el costo de su manipulación. En la figura 3 se muestran conglomerados, cápsulas y nodos. Los objetos de ingeniería en un conglomerado pueden ser conjuntamente *verificados por punto de comprobación*, transferidos a un almacenamiento persistente, reactivados o trasladados a otro nodo, juntos. Esta manipulación de conglomerados completos como una sola operación ofrece una forma de gestión de los sistemas basados en objeto de grano muy fino a un costo razonable. Por ejemplo, un sistema de información geográfica podría considerar que los datos relativos a puntos individuales de un mapa son objetos de ingeniería, pero no podría hacer frente al costo que entrañaría dar a cada uno de estos objetos una existencia completamente individual. La comunicación entre objetos de ingeniería en un conglomerado puede optimizarse mucho, pues los objetos se crean juntos, con un mismo lenguaje, y cabe esperar que se mantengan juntos.

En consecuencia, las interacciones dentro de un conglomerado podrían ser soportadas por una simple invocación de un método local, o su equivalente.

Los conglomerados se controlan y las acciones entre ellos se inician por interacción con un objeto de *gestor de conglomerado* asociado.

8.5.2 Canales

Cuando interactúan objetos de ingeniería en conglomerados diferentes, es necesario un grado bastante alto de mecanismo de soporte. Aunque lo usual es que los objetos estén dentro de la misma cápsula o nodo, se necesitan mecanismos para prever la posibilidad de que uno u otro de ellos termine, falle o se traslade a otro lugar. El conjunto de mecanismos necesarios para realizar esto constituye un canal, que se compone de un número de objetos de ingeniería que interactúan (véase la figura 4).

Los objetos de ingeniería dentro de un canal se dividen en tres tipos según el trabajo que deban realizar. Los *stubs* se ocupan de la información transportada en una interacción, los *vinculadores* se ocupan de mantener la asociación en el interior del conjunto de objetos de ingeniería básicos enlazados por el canal, y los *objetos de protocolo* gestionan la comunicación real.

Los *stubs* interactúan directamente con los objetos de ingeniería básicos soportados por ellos, y realizan funciones como las de activación y desactivación de parámetros, o el registro cronológico de información sobre la interacción que se está realizando. Por tanto, los *stubs* necesitan ganar acceso a la información sobre el tipo de interacción o, de una manera más general, al tipo de interfaz que está siendo soportado. Esto los distingue de los objetos de vinculador y protocolo, que transfieren mensajes completos sin ocuparse de su estructura interna.

Según el diseño del sistema, un *stub* puede ser asociado directamente con un determinado objeto de ingeniería básico o puede ser compartido entre varios de esos objetos. La compartición implica generalmente la necesidad de transferir alguna información adicional para identificar, y de ese modo distinguir, los objetos que están siendo soportados.

Los *vinculadores* se encargan de resolver muchos de los problemas de distribución. El *vinculador* establece la vinculación cuando se crea el canal y subsiguientemente mantiene la integridad de extremo a extremo del canal. Esto significa que trata los cambios de configuración y los fallos de comunicación de los objetos, y mantiene el rastro de otros puntos extremos si unos objetos se han desplazado, o han fallado y han sido reemplazados (proceso de reubicación de objeto). Los *vinculadores* intervienen por tanto en la provisión de muchas de las transparencias de distribución.

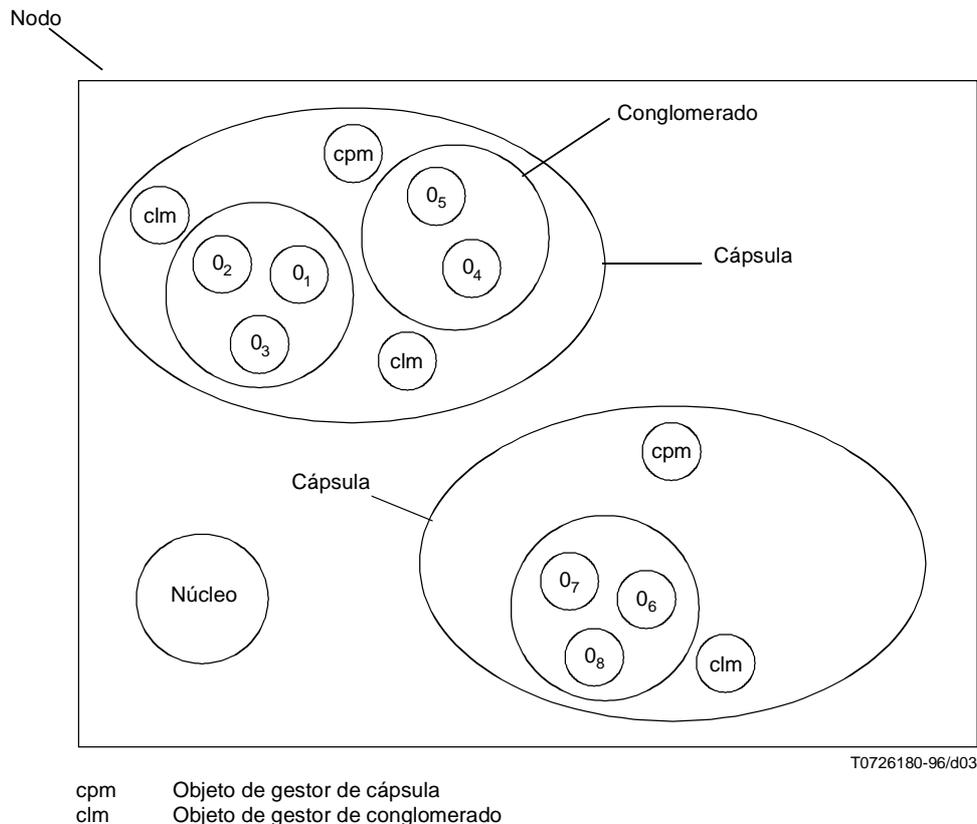
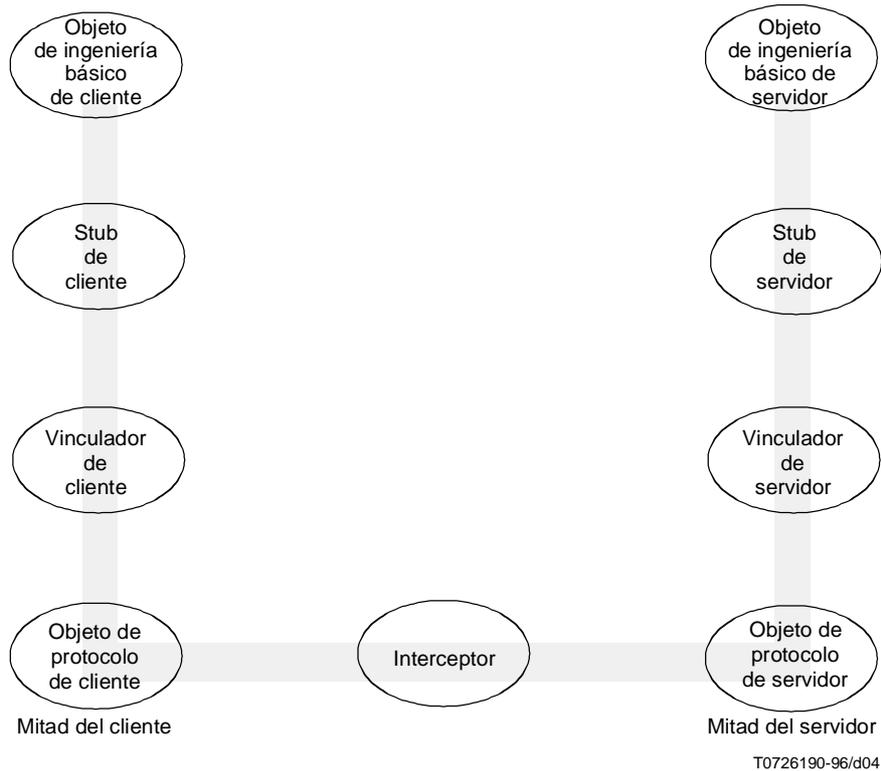


Figura 3 – Cápsulas, conglomerados y nodos



**Figura 4 – Ejemplo de un canal cliente-servidor**

Los objetos de protocolo proporcionan una comunicación de calidad y fiabilidad suficientes entre los vinculadores a los que ellos sirven. Además de manipular cualesquiera protocolos pares que se estén utilizando, los objetos de protocolo dan acceso a servicios de soporte, como los servicios de directorio para el traslado de direcciones, cuando sean necesarios.

Cualquiera de estas tres clases de objetos de ingeniería puede tener necesidad de comunicarse con otras partes del sistema para obtener la información necesaria para su trabajo, o suministrar *información de gestión* a otros objetos de ingeniería. Esta comunicación puede a su vez necesitar las diversas transparencias de distribución, por lo que la comunicación desde estos objetos con cualquier otro lugar se efectúa por medio de un canal; desde este punto de vista, los objetos de ingeniería dentro de un canal desempeñan el papel de objetos de ingeniería básicos en otro. De manera similar cualquiera de estos objetos puede soportar interfaces de control, a través de los cuales pueden ser gestionados. Por ejemplo, un objeto de protocolo puede proporcionar una interfaz de control a través de la cual puede ajustarse la calidad de servicio deseada para el canal.

En aquellos casos en que el canal rebase algún límite técnico u organizacional, puede ser necesario realizar comprobaciones o transformaciones adicionales para que concuerden los requisitos en ambos lados. Estas funciones se realizan por interceptores (descritos más adelante en 8.5.7), que forman parte del canal. Los interceptores pueden tener que efectuar conversiones de formato o de protocolo, o pueden proporcionar comprobaciones de contabilidad o de control de acceso. Un interceptor puede estar constituido por objetos de protocolo, vinculadores y stubs lo que dependerá de la naturaleza del trabajo que se les haya confiado.

Para facilitar su comprensión, en la figura 4 se muestra una configuración de stubs, vinculadores y objetos de protocolo y un interceptor que soporta un canal único entre dos objetos de ingeniería básicos. Sin embargo, en general, esta configuración podría soportar canales entre múltiples pares de objetos de ingeniería básicos (véase la figura 5) o canales con muchos puntos extremos, que soportan diversas formas de comunicación de grupo (véase la figura 6) o la multidifusión. En este último caso, los vinculadores son responsables de la coordinación de la comunicación, pero el mecanismo de multidifusión puede ser proporcionado por objetos de vinculadores o protocolo, lo que dependerá de la tecnología disponible. Se utilizan canales de múltiples puntos extremos para soportar la transparencia de replicación.

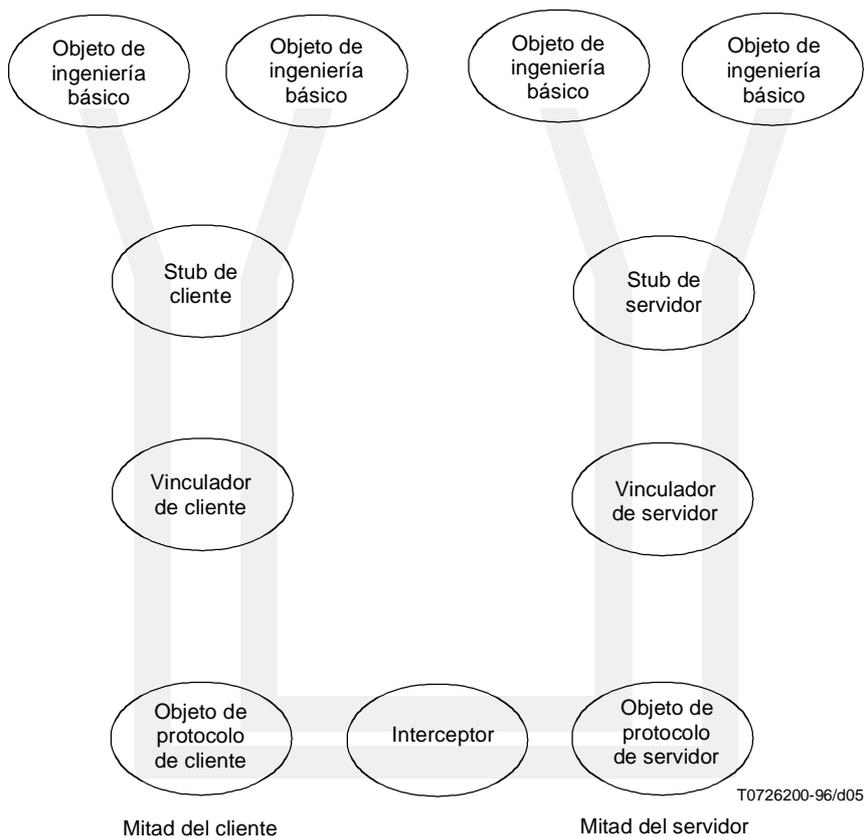


Figura 5 – Ejemplo de una configuración de múltiples canales

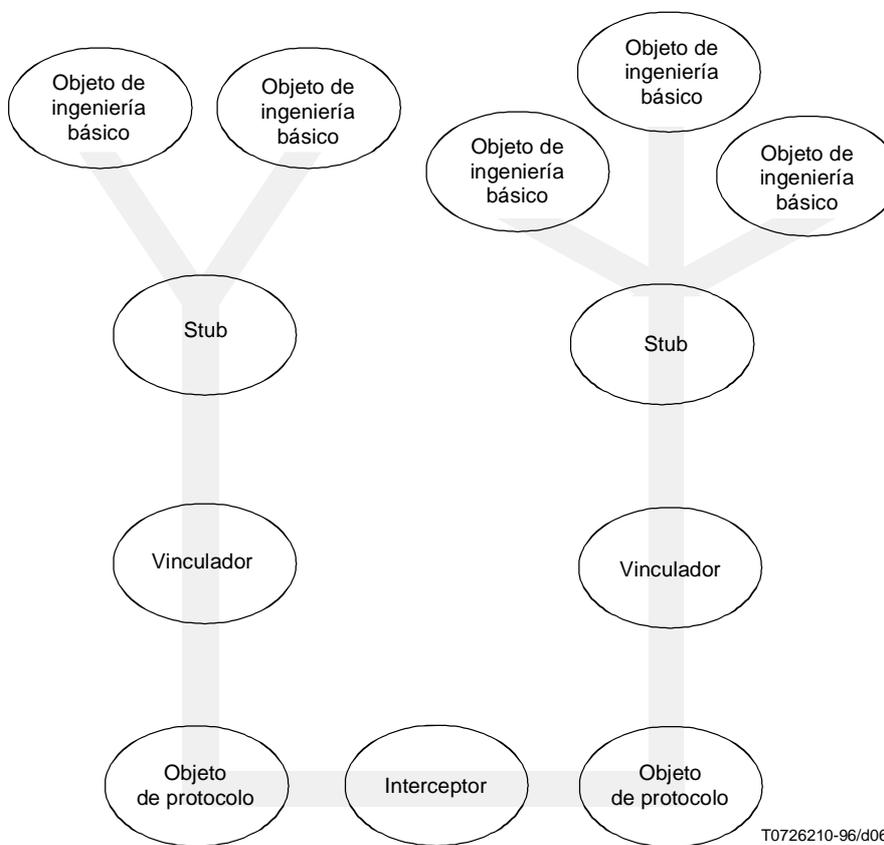


Figura 6 – Ejemplo de una configuración de canal de grupo

### 8.5.3 Referencias de interfaz

Cuando se crea una interfaz, se genera una referencia de interfaz para ella. En este proceso interviene el núcleo, para garantizar que la referencia será inequívoca, y se asignarán e inicializarán recursos suficientes para que los objetos de ingeniería en ese nodo participen en vinculaciones si se les pide que lo hagan.

La referencia de interfaz es el factor esencial para el acceso a una gran cantidad de información. Cuando se ha dado una referencia, es posible descubrir el tipo de la interfaz, una dirección de comunicaciones en la que puede iniciarse la vinculación, y otras informaciones sobre el comportamiento esperado de los stubs, vinculadores y objetos de protocolo dentro del canal, informaciones que son necesarias para que una vinculación ulterior tenga éxito. Es también el punto de partida para invocar las funciones necesarias para el tratamiento de errores; el conocimiento de una referencia de interfaz hace posible establecer contacto con un *reubicador* apropiado.

Esto no implica, sin embargo, que la totalidad de la información se codifique como parte de la referencia de interfaz; si se hiciera esto, el volumen de información sería muy grande y por tanto muy difícil de manipular. El requisito arquitectural es que debe haber alguna prescripción para obtener la información necesaria, a partir de la referencia de interfaz, pero la prescripción exacta, en términos de decodificación e indagación desde otros objetos de ingeniería puede elegirse de manera diferente en diseños de sistemas diferentes.

Además de estas variaciones del diseño, pueden también surgir variaciones por el hecho de que existan múltiples dominios de denominación y se asignen referencias con respecto a estos dominios. Por estas dos razones, puede ser necesario que los interceptores, u otros objetos de ingeniería en el canal, transformen las referencias de interfaz cuando éstas tienen que atravesar varias fronteras de dominios.

### 8.5.4 Vinculación

Hay dos clases de vinculación de ingeniería. Dentro de un conglomerado, o entre los objetos de ingeniería que cooperan dentro de un nodo para proporcionar un canal, hay vinculaciones locales, que son proporcionadas por mecanismos específicos del sistema. Estas vinculaciones se consideran vinculaciones primitivas en la arquitectura. Por otra parte, las vinculaciones soportadas por canales proporcionan transparencias de distribución apropiadas; éstas se denominan vinculaciones distribuidas, y su creación generalmente implicará alguna interacción entre un número de nodos para establecer el canal.

### 8.5.5 Establecimiento de canal

Para establecer una vinculación entre objetos de ingeniería es necesario poder identificar y describir las interfaces de esos objetos. Se utiliza una referencia de interfaz para identificar y describir una interfaz de ingeniería lo suficientemente en detalle para hacer posible que esa interfaz sea ligada. Una referencia de interfaz puede pasarse mediante una o más interacciones entre objetos de ingeniería. Una referencia de interfaz permite, a cualquier objeto que la reciba, iniciar una vinculación con esa interfaz sin ninguna otra información adicional.

Un objeto de ingeniería que desea pasar una referencia de interfaz a una de sus interfaces pide a su núcleo que cree una referencia de interfaz (mediante una operación en su interfaz de gestión de nodo). Una referencia de interfaz identifica la interfaz y proporciona la información necesaria para crear una vinculación con ella.

El tipo de una interfaz de ingeniería conlleva el tipo de interfaz computacional que él implementa, además de la configuración de stubs, vinculadores y objetos de protocolo, y de interceptores (descritos más adelante en 8.5.7) necesarios para soportarlo. Cuando se ligan interfaces de ingeniería con tipos de interfaces computacionales compatibles, en el curso del establecimiento de la vinculación se negocia la configuración exacta del canal necesario para soportar todas las interfaces que comprende. Pudiera ser posible optimizar esta configuración. Por ejemplo, si las interfaces soportan la misma representación de datos, no es necesario configurar el canal con objetos de ingeniería para efectuar conversiones entre las representaciones de datos, aunque será necesario, de todas formas, insertar en un mensaje una copia de los argumentos o resultados.

Las vinculaciones entre interfaces se crean mediante la interacción de objetos de núcleo (por lo general estos son los nodos en los que están ubicadas las interfaces). En consecuencia, cuando un núcleo crea una referencia de interfaz, suministra instrucciones suficientes sobre el modo de hacer contacto con él para establecer una vinculación con la interfaz referenciada. En términos de lenguaje de ingeniería, el núcleo identifica la *interfaz o interfaces de comunicación* en las que tienen que producirse la interacción núcleo-núcleo. La identificación de una interfaz de comunicación podría requerir información sobre los protocolos de comunicación soportados en esa interfaz de comunicación.

Típicamente, el núcleo designa una de sus propias interfaces de comunicación para las interacciones vinculación-establecimiento, pero se permite utilizar también un método más general. La interfaz de comunicación (y su protocolo de comunicación) utilizada en la interacción vinculación-establecimiento no tiene ninguna implicación sobre la interfaz de comunicación (y el protocolo de comunicación) utilizada para soportar la vinculación, una vez que ésta haya sido creada. Un ejemplo de una situación análoga es aquella en que se hace una llamada telefónica para pedir el envío de unos documentos (escritos en papel) por medio del sistema postal.

La referencia de interfaz contiene una información sumamente compleja que puede ser interpretada a todo lo largo del sistema ODP. La estructura detallada de una referencia de interfaz está por tanto sujeta a una normalización aparte. El RM-ODP no prescribe si la referencia de interfaz contiene físicamente la información aquí descrita o si, simplemente, es una clave utilizada para ganar acceso a esta información mediante una interacción con otros objetos de ingeniería (por ejemplo, un nombre, que deba ser resuelto por un servidor de nombres).

Como ejemplo de establecimiento de canal, considérese el establecimiento de un canal de tren entre dos objetos de ingeniería. Esto se efectúa en varios pasos.

**\* Paso 1**

Uno de los dos objetos de ingeniería inicia la configuración de un canal mediante interacción con su núcleo. La sintaxis de interacción puede tener el formato siguiente:

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

donde *StreamChannel* es el tipo "*StreamChannel*" y *Streamchannel* es el tipo del canal que ha de crearse; *producer/consumer* indica que el objeto de ingeniería en cuestión tendrá el rol de productor y el rol de consumidor para el canal de tren; *IFPC1* es la interfaz del objeto de ingeniería que habrá de ligarse al canal de tren.

Cuando se produce esta interacción, el núcleo crea un objeto de stub, un objeto de vinculación y un objeto de protocolo que corresponden al tipo y al rol del canal. Estos objetos de ingeniería se ligan para crear una primera parte de un canal de tren. La interfaz de presentación del objeto de stub se liga a la interfaz *IFPC1*. El objeto de stub se liga entonces al objeto de vinculación que está ligado al objeto de protocolo. El resultado de esta interacción es una referencia de interfaz (*IFrefStreamchannel*). La referencia de interfaz se comunicará a los objetos de ingeniería que desean vincularse al canal.

**\* Paso 2**

La referencia de interfaz del canal se comunica al segundo objeto de ingeniería. Este objeto interactúa con su núcleo para vincularse al canal por medio de la siguiente interacción:

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

donde *StreamChannel* es el tipo de canal; *producer/consumer* indica que el segundo objeto de ingeniería desempeñará los papeles de productor y de consumidor para el canal de tren; *IFPC2* es la interfaz del segundo objeto que habrá de vincularse al canal de tren.

Basándose en la referencia de canal *IFrefStreamchannel*, el núcleo determina el tipo de canal y la ubicación de los objetos de protocolo para los otros participantes en el canal de tren. El núcleo crea un objeto de stub, un objeto de vinculación y un objeto de protocolo que corresponden al tipo de canal y al rol de los otros participantes. Estos objetos se vinculan a la parte del canal de tren ya establecida y al segundo objeto de ingeniería. Entonces, los vinculadores en el canal interactúan unos con otros para permitir la comunicación a través del canal.

**\* Paso 3**

Otros objetos pueden vincularse al canal existente utilizando la misma interacción **BindChannel** ().

### 8.5.6 Interfaces de gestión

Solamente un objeto puede modificar su propio comportamiento. Un objeto puede responder a peticiones de una aplicación de gestión para que modifique su comportamiento y puede por consecuencia delegar, a la aplicación de gestión, la responsabilidad sobre alguna parte de su gestión.

La gestión de recursos requiere que sea posible invocar operaciones de gestión sobre servicios individuales, los objetos de ingeniería que los contienen, el conglomerado que contiene al objeto de ingeniería, la cápsula que contiene al conglomerado y el nodo que soporta la cápsula.

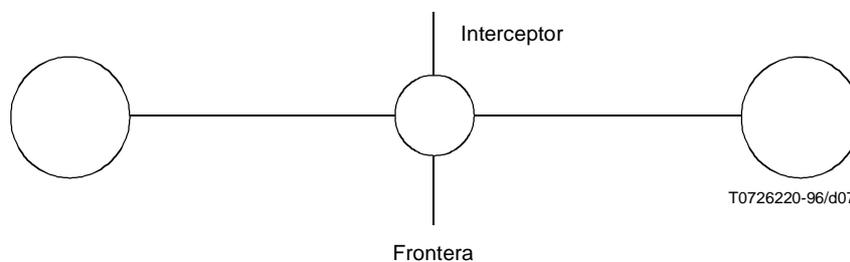
### 8.5.7 Interceptores

Se especifican interceptores para satisfacer requisitos, identificados en una especificación de empresa, para la federación de dominios de tecnología o administrativos. Los interceptores corresponden a las nociones de objetos de "cabecera", "agentes", o "monitor", que están situados entre dos dominios y habilitan o permiten interacciones sobre la base de un contrato entre las Administraciones que especifica la base para su federación.

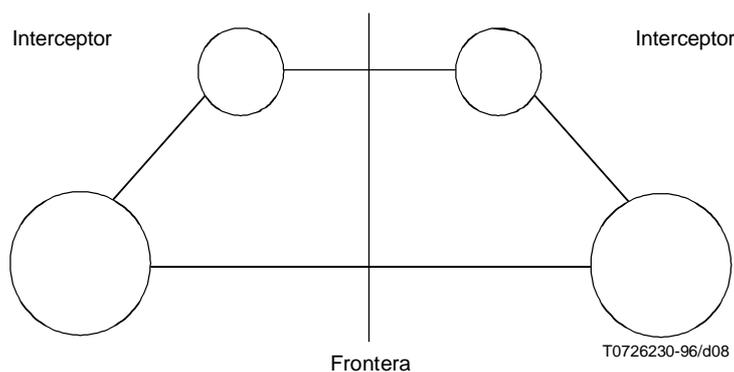
Dentro de un dominio de tecnología, los objetos de núcleo tienen idénticas representaciones de datos y funcionalidad de protocolos, denominación y direccionamiento. Cuando dos dominios de tecnología se encuentran, existe una oportunidad para fusionarlos, es decir, los objetos computacionales de cada dominio son ampliados para que utilicen las tecnologías de ambos dominios, es decir, del propio y del ajeno (o para que ambos conmuten a una tecnología común más genérica). Cuando esto ocurre, la frontera entre las tecnologías desaparece, y basta con la transparencia de acceso. Los interceptores entran en juego en el caso de que no sea posible modificar la tecnología de uno o ambos dominios y, en consecuencia, la interceptación debe producirse en la frontera, proporcionando conversión de protocolos y traducción de nombres.

La traducción de protocolos y datos a través de las fronteras de tecnologías se efectúa por interceptores en línea, y estos intervienen en todas las interacciones de objetos a través de la frontera. Para una mayor eficiencia, sólo se utilizaría un interceptor en línea para cada frontera de tecnología (véase la figura 7).

Los interceptores de frontera administrativa existen enteramente dentro de una Administración y ejercen las responsabilidades de protección para una Administración. Por ejemplo, un interceptor de frontera administrativa podría utilizarse en la traducción de información de seguridad que contiene permisos. Tal interceptor podría utilizarse antes de una interacción por la infraestructura de objeto computacional invocante. Un interceptor administrativo puede comunicar con el interceptor similar en el otro dominio para intercambiar información como claves criptográficas, y para comprobar la información administrativa antes de traducirla (véase la figura 8). Las traducciones efectuadas por interceptores administrativos pueden utilizarse en varias interacciones sucesivas entre objetos computacionales sin una ulterior utilización del interceptor.



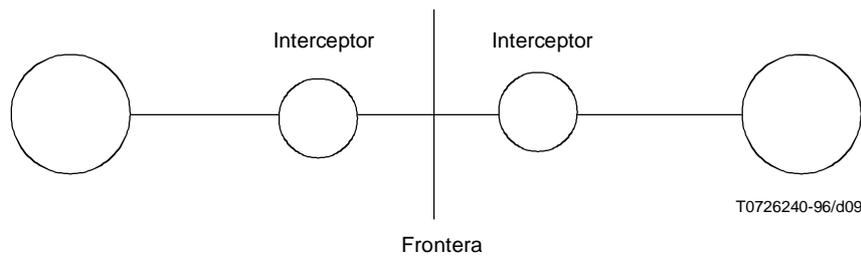
**Figura 7 – Interceptor en línea – Frontera tecnología**



**Figura 8 – Interceptor dividido – Frontera administrativa**

Cuando coexisten una frontera administrativa y una frontera de tecnología, se utilizarían interceptores en línea y fuera de línea (una combinación de las figuras 7 y 8). Como otra posible solución, se puede dividir el interceptor en línea de manera que haya dos interceptores en línea, uno en cada dominio administrativo, para soportar las relaciones de trust (véase la figura 9).

En general, las fronteras son N-direccionales porque más de dos subsistemas pueden encontrarse en la misma ubicación; lógicamente, sin embargo, una frontera N-direccional puede modelarse siempre como N fronteras bidireccionales y, de ese modo, en lo tocante a la arquitectura, sólo hay que considerar el caso del interceptor bidireccional.



**Figura 9 – Interceptor dividido – Frontera combinada de tecnología y administrativa**

Se presentan condiciones especiales cuando la federación requiere que los comerciantes en cada dominio estén enlazados. Si un comerciante es accesible a través de un interceptor en línea, dicho comerciante puede utilizarse para ganar acceso a comerciantes subsiguientes, pero el interceptor tiene que proporcionar funciones de inicialización especiales para permitir el acceso a por lo menos un comerciante inicial, a través de sí mismo, en ambos sentidos de transmisión. Esta inicialización puede obtenerse haciendo que el interceptor entre en federación con comerciantes a ambos lados de su frontera, haciendo que el interceptor se convierta en un apoderado (o representante) del comerciante o incorporando un comerciante en el propio interceptor.

### 8.5.8 Puntos de conformidad

La estructuración de la especificación de ingeniería en conglomerados, cápsulas y nodos, y el soporte de la interacción por canales estructurados da lugar a un gran número de interfaces, cualquiera de las cuales puede seleccionarse como un punto de conformidad, lo que permite observaciones y pruebas de conformidad.

Las diversas interfaces implican diferentes clases de conformidad. La interfaz entre objetos de protocolo es un punto de conformidad de interfuncionamiento, que proporciona métodos usuales, como las prueba OSI, basados en la observación del comportamiento de la comunicación. La mayor parte de las otras interfaces son internas a un nodo, y representan fronteras entre módulos de soporte lógico; son *puntos de referencia programática* y permiten efectuar pruebas para determinar la compatibilidad y la portabilidad del soporte lógico. Algunas de las interfaces con objetos de ingeniería básicos pueden permitir otras formas de pruebas de conformidad, para el intercambio o conformidad perceptual (interacción correcta con el mundo real); asimismo, pueden ser puntos de conformidad en los que se evalúa el comportamiento para asegurar la consistencia con los requisitos indicados en las especificaciones de empresa y de información.

## 8.6 Lenguaje de tecnología

La especificación de tecnología describe la implementación del sistema ODP en términos de una configuración de objetos de tecnología que representan los componentes de soporte físico y de soporte lógico de la implementación. Está constreñida por el costo y la disponibilidad de los objetos de tecnología (productos de soporte físico y de soporte lógico) que satisfarían esta especificación. Éstos pueden ser conformes con normas que puedan ser implementadas y que son efectivamente plantillas para objetos de tecnología. Así, el punto de vista de tecnología proporciona un enlace entre el conjunto de especificaciones de puntos de vista y la implementación real, indicando las normas utilizadas para proporcionar las operaciones básicas necesarias en las otras especificaciones de punto de vista, y la especificación de tecnología tiene por finalidad proporcionar la información suplementaria necesaria para la implementación y las pruebas seleccionando soluciones normalizadas para componentes básicos y mecanismos de comunicación. Tal selección es necesaria para completar la especificación del sistema, pero se aparta bastante del resto del proceso de diseño.

La selección de la tecnología tiene consecuencias, sin embargo. Un área en la que las selecciones en la especificación de tecnología repercuten en otros aspectos del diseño del sistema es la provisión de una determinada calidad de servicio. Las selecciones en el punto de vista de la tecnología determinan los costos de las interacciones en lo que respecta a la calidad de funcionamiento y, por tanto, indirectamente, la calidad de servicio que puede obtenerse por el comportamiento definido en otras especificaciones de punto de vista.

La especificación de tecnología desempeña un papel importante en el proceso de las pruebas de conformidad. Esta especificación identifica los puntos de conformidad en el sistema real, en los que un probador puede hacer observaciones de su comportamiento y suministra la información necesaria para interpretar las observaciones que un probador puede hacer en términos del vocabulario y los conceptos utilizados en los otros puntos de vista de las especificaciones del sistema. Por ejemplo, permite reconocer interacciones válidas, de manera que se pueda verificar, con respecto a algún comportamiento especificado de un objeto de tecnología, la medida en que dichas interacciones son apropiadas. La información requerida para esta finalidad se denomina información suplementaria de implementación para las pruebas (IXIT, *implementation extra information for testing*).

## 8.7 Consistencia entre los puntos de vista

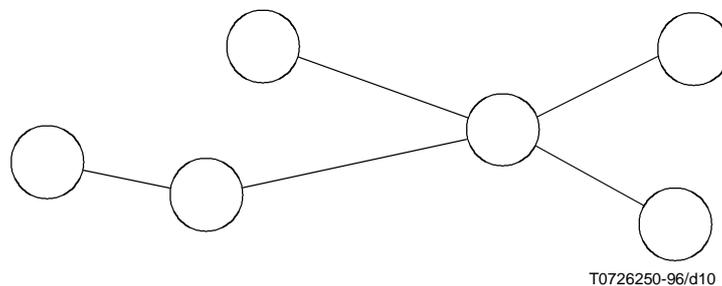
Las cinco especificaciones de punto de vista para un sistema están enlazadas por enunciados que definen las relaciones entre los términos esenciales contenidos en ellas y establecen que:

- las especificaciones se relacionan con un objeto único y no son independientes;
- las especificaciones son consistentes consigo mismas;
- el comportamiento observable en puntos de conformidad en la especificación de tecnología puede relacionarse con requisitos en las otras especificaciones de punto de vista.

Muchos de los enlaces que se necesitan serán proporcionados implícitamente por las notaciones utilizadas, resultantes de correspondencias entre nombres. Sin embargo, algunas de las constricciones esenciales deberán enunciarse explícitamente. En la arquitectura se imponen constricciones a las relaciones entre términos en los propios lenguajes de punto de vista, estableciendo algunos límites sobre las correspondencias que pueden establecerse. La mayor parte de las constricciones se imponen entre términos en los lenguajes computacionales y de ingeniería, y se definen de modo que creen interpretaciones consistentes cuando se especifican separadamente componentes de sistema, como los que soportan las funciones ODP.

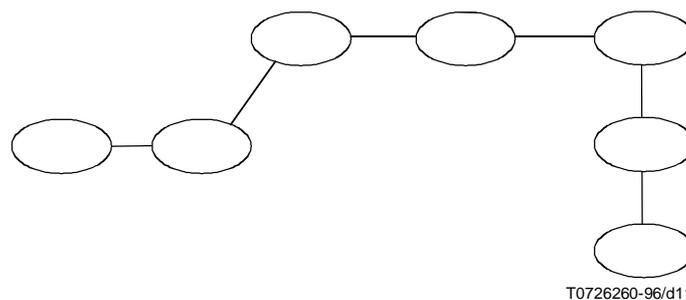
Se necesitan correspondencias claras entre puntos de vista si los procesos de identificación de interfaces y de provisión de transparencias deben ser soportados automáticamente por herramientas de desarrollo. Por ejemplo, un objeto computacional puede realizarse como un conjunto de objetos de ingeniería enlazados, pero un objeto de ingeniería único no puede representar múltiples objetos computacionales; una interfaz computacional no puede dividirse en interfaces de ingeniería distintas, excepto donde estén relacionadas por funciones de replicación; las interfaces computacionales se identifican inequívocamente mediante identificadores de ingeniería. Estas clases de construcción ayudan a asegurar que los mecanismos comunes de ingeniería serán capaces de soportar la gama completa de comportamientos computacionales posibles.

A fin de ilustrar la naturaleza de las correspondencias que pueden aplicarse, supóngase, por ejemplo, que cierta especificación de un sistema dado puede representarse como un conjunto de objetos de punto de vista 1 (V1), como los representados en la figura 10.



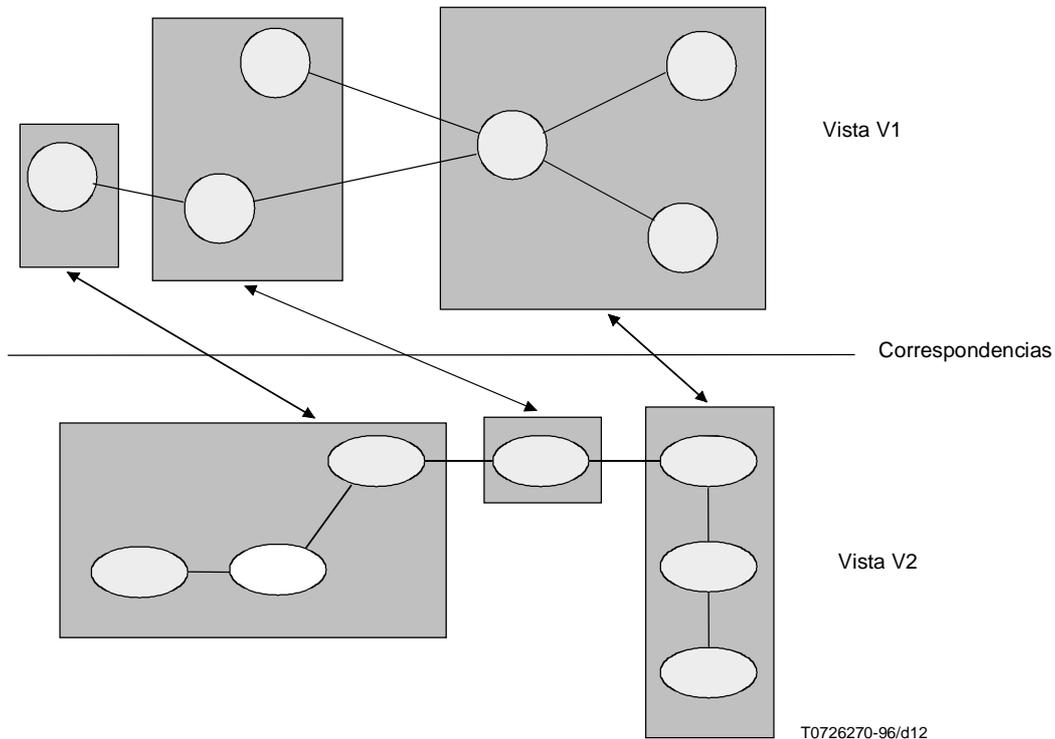
**Figura 10 – Una vista V1 de un sistema**

El mismo sistema podría describirse desde otro punto de vista (V2) como un conjunto diferente de objetos V2 que interactúan, como se muestra en la figura 11.



**Figura 11 – Una vista V2 de un sistema**

Puesto que se trata de dos descripciones del mismo sistema, es posible agrupar los objetos indicados en las dos figuras precedentes, con el fin de verificar que las reglas de correspondencia se satisfacen entre los grupos. Este proceso se muestra en la figura 12.



**Figura 12 – Correspondencias entre diferentes puntos de vista de un sistema**

Si tal correspondencia no puede establecerse, las dos descripciones diferentes no son consistentes y deben ser refinadas hasta que pueda demostrarse una correspondencia.

Las configuraciones de objetos que se comparan (las configuraciones de casillas en la figura 12) se definen, en general, con la sola finalidad de hallar una correspondencia entre dos especificaciones.

En otras palabras, para comparar una especificación SpecA escrita en un lenguaje de punto de vista dado L1, con otra especificación SpecC del mismo sistema, escrito en otro lenguaje de punto de vista L2, en el caso general es necesario:

- a) Transformar SpecA en otra especificación en L2. Denomínese esta especificación SpecB. Obsérvese que el RM-ODP **no define** ningún algoritmo de transformación.

NOTA – A veces puede ser conveniente, para llevar a cabo esta transformación, derivar a partir de SpecA otra especificación del mismo sistema, también escrita en L1, que sea equivalente a SpecA, para verificar mejor las correspondencias. Esto corresponde, por ejemplo, a agrupar los objetos definidos en el sistema de la figura 12 en otros objetos (las casillas).

- b) Verificar que no hay contradicciones entre SpecB y SpecC.

Las correspondencias son aplicables entre especificaciones expresadas en diferentes lenguajes de punto de vista, no entre términos de esos lenguajes. En otras palabras, no hay una traducción directa de un lenguaje de punto de vista a otro.

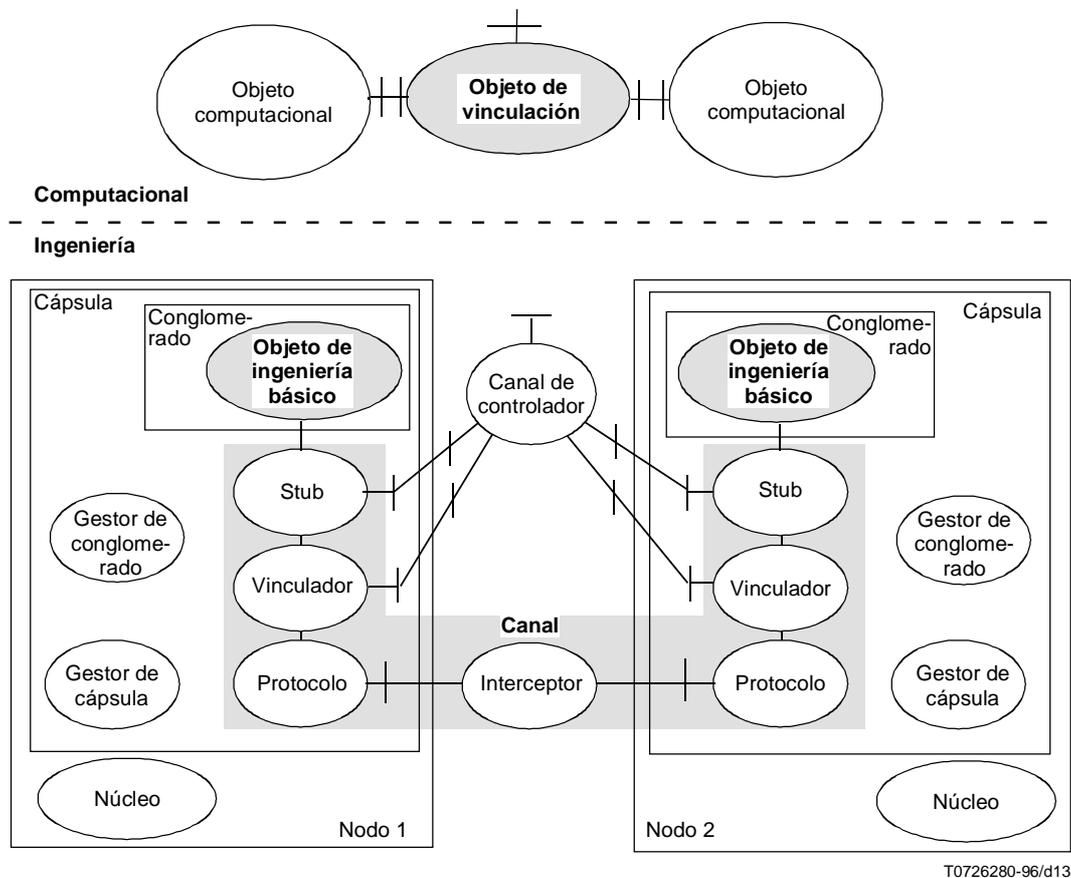
### 8.7.1 Consistencia del punto de vista de la empresa con otros puntos de vista

El lenguaje de empresa debe servir de base para especificar metas de las empresas que deben reflejarse directamente o indirectamente en todas las otras especificaciones de punto de vista. El punto de vista de la empresa describe, explícitamente, los objetivos del sistema en el contexto de la organización, en términos de miembros, roles, acciones, finalidades, utilización y políticas.

Por consiguiente, una especificación de punto de vista de la información, computacional, de ingeniería o de tecnología es consistente con una especificación de empresa si todos los roles, actividades y políticas descritas en la especificación de empresa se reflejan correctamente. Por ejemplo, los esquemas dinámicos definidos en una especificación de información deben obedecer a las políticas descritas en la especificación de empresa. Roles diferentes identificados en la especificación de empresa pueden ser soportados por objetos computacionales que tengan diferentes requisitos de transparencia. Así, las necesidades de transparencia para cada rol en la especificación de empresa deben reflejarse mediante el empleo del correspondiente mecanismo de transparencia en la especificación de ingeniería. Un requisito de flexibilidad o una política en la especificación de la empresa pueden conducir a la elección de tecnologías específicas para la implementación de un sistema distribuido.

### 8.7.2 Correspondencias entre especificaciones computacionales y de ingeniería

Existe una correspondencia entre la especificación computacional y una especificación de ingeniería que va ejecutarse. Esta especificación de ingeniería muestra el comportamiento descrito en la especificación computacional. La figura 13 es un ejemplo de esta correspondencia.



**Figura 13 – Ejemplo de correspondencia entre puntos de vista computacionales y de ingeniería**

Los objetos de ingeniería básicos corresponden a objetos computacionales. Los objetos de ingeniería básicos se agrupan en conglomerados que pudieran representar, por ejemplo, piezas ejecutables de código escritos en los lenguajes de programación C o C++. Los conglomerados se organizan en una cápsula que podría representar, por ejemplo, un proceso UNIX. La cápsula está ligada a un objeto de núcleo (que representa, por ejemplo, un determinado sistema operativo) perteneciente a un nodo (que representa, por ejemplo, una estación de trabajo). Los objetos de ingeniería básicos son soportados por objetos de soporte de ingeniería adicionales como se muestra en la figura 13.

El refinamiento de plantillas computacionales para formar plantillas de ingeniería corresponden a la noción de *compilar* programas para producir código de objeto. El refinamiento de plantillas de ingeniería para formar *plantillas de conglomerado* corresponde a la noción de *enlazar (linking)* módulos para formar la imagen de un programa ejecutable. El concepto de cápsula corresponde a la noción de *espacio de dirección o proceso* en la mayoría de los sistemas operativos.

El objeto de vinculación en la especificación computacional representado en la figura 13 corresponde a una configuración de canal en la especificación de ingeniería. Las constricciones de entorno específicas de las interfaces que están siendo vinculadas (por ejemplo, las relativas a la seguridad, calidad de servicio) se tienen en cuenta al establecer un canal entre los objetos de ingeniería básicos que intervienen.

La interfaz de control del objeto de vinculación en la especificación computacional corresponde a interfaces con stubs y vinculadores en nodos diferentes, en la representación de ingeniería. Se podría introducir un objeto de controlador de canal que tuviese a su cargo el despacho de operaciones de control. La comunicación entre el objeto de controlador de canal y el stub y los vinculadores se efectúa mediante canales establecidos para esta finalidad (que no se muestran en la figura 13). Se puede crear objetos de ingeniería de soporte (por ejemplo, objetos de sincronización) para gestionar y controlar un conjunto de canales conexos.

De una manera más general, una especificación computacional describe la funcionalidad de un sistema como un conjunto de objetos computacionales que interactúan. La especificación de ingeniería está constreñida por la especificación computacional, ya que tiene que respetar los objetos computacionales y sus interfaces. Cada objeto computacional tiene que estar representado como un objeto de ingeniería básico o como un grupo de objetos de ingeniería básicos. En el caso más simple, un objeto de ingeniería único es equivalente a un objeto computacional. Por ejemplo, un programa de aplicación, una vez compilado, corresponde a un objeto de ingeniería que habrá de cargarse como un módulo de carga en un conglomerado después del enlace-edición con otros objetos como stub, vinculadores, etc. Esta configuración se representa en la figura 14.

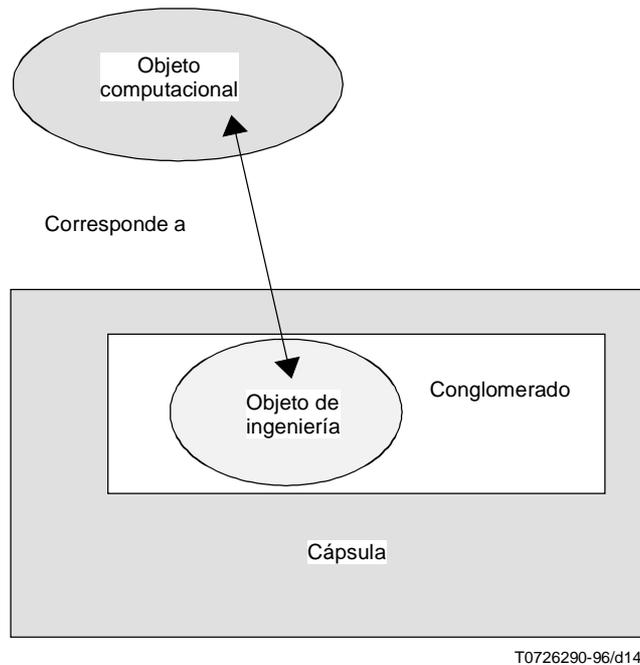
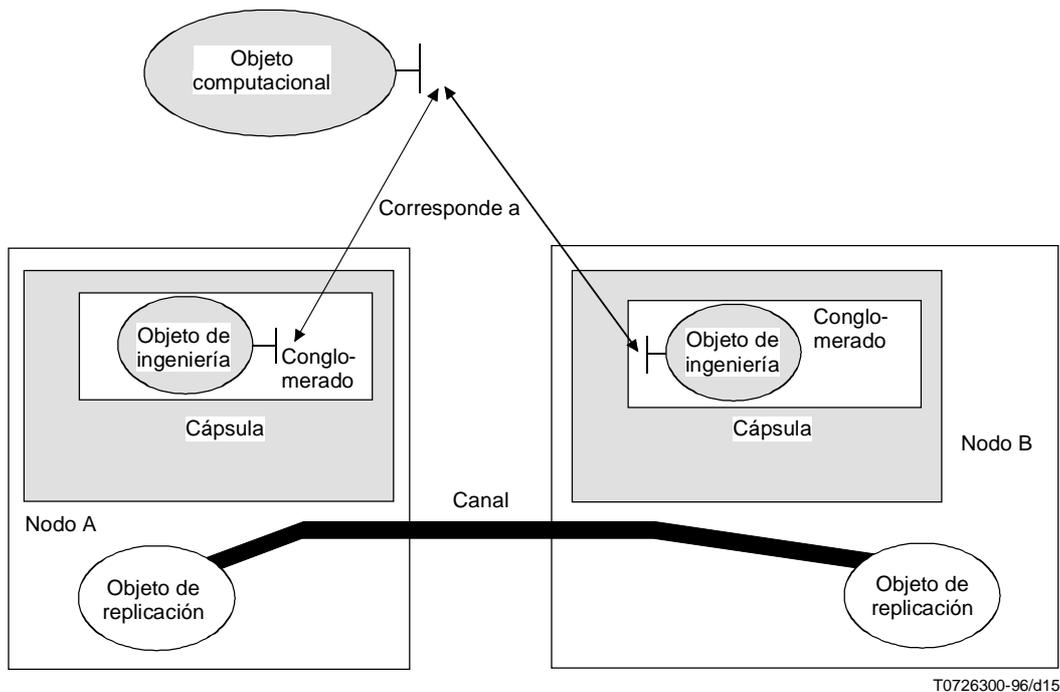


Figura 14 – Correspondencia de uno a uno

Para cada interfaz computacional tiene que haber una interfaz de ingeniería correspondiente, a menos que intervengan transparencias que repliquen objetos. En este caso, una misma interfaz computacional puede estar asociada con diferentes identificadores de interfaz de ingeniería, lo que permite la replicación, por ejemplo por razones de calidad de funcionamiento. La interfaz computacional va asociada con un conjunto de *referencias de interfaces de ingeniería*, que corresponden a diferentes objetos de ingeniería. Las actividades de estos objetos de ingeniería tienen que ser coordinadas por objetos de replicación para asegurar que el sistema mantiene un estado global consistente. En la figura 15 se presenta un ejemplo de este caso, que muestra dos objetos de ingeniería básicos, ubicados en dos nodos diferentes, que replican las funcionalidades ofrecidas en una interfaz computacional. Como los objetos de ingeniería están en nodos diferentes, la coordinación de la replicación la aseguran dos objetos de replicación, uno en cada nodo, que se comunican a través de un canal.



**Figura 15 – Correspondencia de muchos a uno**

El proceso de refinamiento entre la especificación computacional y la especificación de ingeniería puede consistir, simplemente, en la identificación de objetos de soporte adecuados para que figuren en los canales que representan objetos de vinculación en la especificación computacional. En otros casos, es posible que las plantillas para los objetos computacionales propiamente dichos deban sufrir una transformación considerable, reemplazándose los enunciados declarativos sobre comportamiento (por ejemplo, constricciones de sincronización) por una utilización explícita de funciones ODP apropiadas (por ejemplo, la función de transacción).

Si se tiene un conocimiento suficiente de la configuración del sistema, pueden obtenerse optimizaciones particulares en aquellos casos en que no se requiere que la estructura de ingeniería sea completamente general. Así, un implementador puede contornear interacciones entre objetos de ingeniería básicos en el mismo nodo (por ejemplo, utilizando llamadas de procedimientos locales para mejorar la calidad de funcionamiento) siempre que dichos contorneos no afecten al interfuncionamiento a través de las interfaces visibles por los objetos en otros núcleos. De esta forma, la arquitectura atiende a la distribución sin tener necesidad de que una multiplicidad de mecanismos se encarguen de interacciones locales y distantes, y, al mismo tiempo, sin sacrificar la eficiencia de la ejecución.

## 8.8 Funciones ODP

La Rec. UIT-T X.903 | ISO/CEI 10746-3 proporciona descripciones generales de un conjunto de funciones ODP. Estas funciones son fundamentales para la construcción de sistemas ODP, o ampliamente aplicables a ella. Las especificaciones detalladas para estas funciones estarán sujetas a actividades de normalización específicas y las normas resultantes pueden combinar especificaciones de funciones ODP para proporcionar especificaciones para componentes de sistemas ODP.

El conjunto completo de funciones se divide en cuatro grupos:

- a) funciones de gestión;
- b) funciones de coordinación;
- c) funciones de depositario;
- d) funciones de seguridad.

### 8.8.1 Funciones de gestión

Las funciones de gestión comprenden:

- la función de gestión de nodo;
- la función de gestión de objeto;

## ISO/CEI 10746-1 : 1997 (S)

- la función de gestión de conglomerado; y
- la función de gestión de cápsula.

La función de gestión de nodo la proporciona el núcleo de un nodo, y se encarga de las funciones de control del procesamiento, el almacenamiento y las comunicaciones en un nodo. Proporciona:

- la gestión de hilos de procesamiento;
- el acceso al reloj y la gestión de temporizadores;
- la creación de canales y la manipulación de referencias de interfaces de ingeniería;
- la instanciación de plantillas de cápsulas y la supresión de cápsulas.

La función de gestión de objetos la proporciona cualquier objeto, cuando se requiera, y permite la verificación por punto de comprobación y la supresión del objeto.

La función de gestión de conglomerado la proporciona un gestor de conglomerado y permite la verificación por punto de comprobación, recuperación, migración, desactivación o supresión del conglomerado.

La función de gestor de cápsula la proporciona un gestor de cápsula y permite la instanciación (incluida la recuperación y reactivación), verificación por punto de comprobación, desactivación o supresión de todos los conglomerados en la cápsula, y la supresión de la propia cápsula.

### 8.8.2 Funciones de coordinación

Las funciones de coordinación comprenden:

- la función de notificación de evento;
- la función de verificación por punto de comprobación y recuperación;
- la función de desactivación y reactivación;
- la función de grupo;
- la función de replicación;
- la función de migración;
- la función de transacción; y
- la función de rastreo de referencia de interfaz de ingeniería.

La función de notificación de eventos registra y pone a disposición historias de eventos. Los productores de eventos interactúan con la función para crear historias de eventos y la función notifica a los consumidores de eventos registrados la disponibilidad de historias de eventos.

La función de verificación por punto de comprobación y recuperación coordina la verificación por punto de comprobación y la recuperación de conglomerados que han fallado, desde los *puntos de comprobación*. Se rige por políticas que indican cuándo deben verificarse por punto de comprobación los conglomerados y dónde deben almacenarse los puntos de comprobación asociados, cuándo y dónde deben recuperarse los conglomerados, y qué punto de recuperación debe recuperarse.

La función de desactivación y reactivación coordina la desactivación y reactivación de conglomerados. Se rige por políticas que indican cuándo deben desactivarse los conglomerados y dónde deben almacenarse los puntos de comprobación asociados, cuándo y dónde deben reactivarse los conglomerados, y qué punto de recuperación debe utilizarse para la reactivación.

La función de grupo proporciona los mecanismos necesarios para coordinar las interacciones de objetos en una vinculación multipartita.

La función de replicación se emplea en el caso especial de un grupo en el que los objetos pertenecientes al mismo son compatibles en comportamiento. Proporciona los mecanismos necesarios para asegurar que el grupo sea percibido por otros objetos como si fuera un objeto único, y permite también aumentar o disminuir el número de miembros del grupo. La función puede utilizarse en el nivel de un conglomerado, junto con la función de grupo, para formar un conjunto combinado de grupos de réplica en el que los objetos de cada conglomerado forman un grupo de réplica.

La función de migración coordina el desplazamiento de un conglomerado de una cápsula a otra. Puede operar por replicación del conglomerado, utilizando la función de replicación, o desactivando el conglomerado y reactivándolo en otro conglomerado, utilizando la función de desactivación y reactivación.

La función de transacción coordina y controla un conjunto de transacciones para alcanzar un nivel especificado de visibilidad y permanencia, con arreglo a las políticas que determinan las acciones de interés para la transacción. La transacción ACID es un caso especial de la función de transacción para el cual las transacciones tienen las propiedades de atomicidad, consistencia, aislamiento y durabilidad.

La función de rastreo de (referencia de) interfaz de ingeniería supervisa la transferencia de referencias de interfaz de ingeniería entre objetos de ingeniería en conglomerados diferentes para determinar el momento en que la infraestructura de soporte para la referencia deja de ser necesaria porque ningún objeto en ningún otro conglomerado puede vincularse con la interfaz referenciada.

### 8.8.3 Funciones de depositario

Las funciones de depositario comprenden:

- la función de almacenamiento;
- una función de organización de información;
- la función de reubicación;
- la función de depositario de tipo; y
- la función de comercio.

La función de almacenamiento almacena datos.

La función de organización de información gestiona un depositario de información descrito por un esquema de información y permite la modificación y actualización del esquema y del depositario, así como interrogar al depositario.

La función de reubicación gestiona un depositario de ubicaciones para interfaces y funciones de gestión para conglomerados que soportan esas interfaces.

El depositario de tipos gestiona un depositario de especificaciones de tipos y de relaciones de tipos.

La función de comercio (*trading function*) soporta la exportación de *ofertas de servicio*, hechas por proveedores de servicios, en forma de información sobre la interfaz en la que se proporciona el servicio y la importación, por usuarios del servicio, de ofertas de servicio que se ajustan a requisitos específicos.

### 8.8.4 Funciones de seguridad

Las funciones de seguridad atienden a los requisitos de confidencialidad, integridad, disponibilidad y responsabilidad de rendición de cuentas ("*accountability*"). Comprenden:

- la función de control de acceso;
- la función de auditoría de seguridad;
- la función de autenticación;
- la función de integridad;
- la función de confidencialidad;
- la función de no repudio; y
- la función de gestión de claves.

La función de control de acceso impide las interacciones no autorizadas con un objeto.

La función de auditoría de seguridad supervisa y capta información sobre las acciones relacionadas con la seguridad, y permite el análisis de la información para revisar las políticas, controles y procedimientos.

La función de autenticación da la seguridad de que la identidad pretendida de un objeto es cierta.

La función de integridad detecta y/o impide la creación, modificación o supresión no autorizadas de datos.

La función de confidencialidad impide la revelación no autorizada de información.

La función de no repudio impide que un objeto que está interviniendo en una relación niegue su participación en la interacción.

La función de gestión de claves proporciona facilidades para la gestión de claves criptográficas.

Las funciones proporcionan servicios que pueden aplicarse tanto a los propios objetos como a las interacciones entre objetos. Los mecanismos requeridos para proporcionar servicios de seguridad deben, ellos mismos, estar protegidos, ya que una amenaza inteligente y maliciosa es característica de los entornos en que se requiere seguridad. Una encapsulación por ingeniería puede ayudar a proporcionar esta protección. En muchos casos pueden proporcionarse servicios de seguridad sin que se requiera una referencia en especificaciones computacionales.

## 8.9 Transparencias de distribución ODP

Como se describe en 8.5, los objetos de ingeniería interactúan unos con otros a través de stubs, vinculadores, objetos de protocolo, interceptores y núcleos. Los objetos de ingeniería cooperan para proporcionar una transparencia al llevar uniformidad a algún aspecto de la distribución de los objetos de ingeniería básicos por ellos soportado. Por ejemplo, pueden traducir invocaciones en intercambios de mensajes utilizando un formato de datos común para enmascarar las diferencias en la codificación de los datos.

Algunas formas de transferencia requieren funciones de soporte. Por ejemplo, si los objetos de ingeniería pueden desplazarse de una ubicación a otra, se requiere un medio de registrar y averiguar la ubicación actual de un componente (función de reubicación).

Las propias funciones de soporte pueden tener necesidades de transparencia. Se puede replicar una función de reubicación para acrecentar su disponibilidad.

Las funciones de soporte se modelan como objetos de ingeniería de modo que la arquitectura proporcione el grado máximo de flexibilidad de configuración y de reutilización de conceptos arquitecturales al definir la distribución de estas funciones. En una implementación se puede, por ejemplo, cobicar funciones de soporte, es decir, situarlas en un mismo lugar, para obtener una interacción eficaz, o replicarlas para lograr una mayor fiabilidad.

Transparencia de distribución es la propiedad que consiste en ocultar las propiedades de distribución, con respecto a los usuarios de extremo y los especificadores, en los lenguajes de empresa, de información y computacional. Las normas de "constitución mediante componentes" contendrán instrucciones precisas para la utilización de funciones y otros componentes de base para proporcionar transparencias.

### 8.9.1 Transparencia de acceso

La transparencia de acceso permite el interfuncionamiento a través de arquitecturas de computador y lenguajes de programación heterogéneos.

La transparencia de acceso es de importancia crítica en la construcción de sistemas distribuidos que utilizan arquitecturas de computador, lenguajes de programación y otros recursos heterogéneos.

### 8.9.2 Transparencia de fallo

La transparencia de fallo *oculta*, con respecto a un objeto computacional, el fallo y la posible recuperación de otros objetos computacionales, o del objeto en cuestión, para permitir una tolerancia a las averías. Puede proporcionarse por una infraestructura adecuada. En otro caso, es soportada por la función de verificación por punto de comprobación y recuperación, o por la función de replicación, junto con la función de reubicación.

Un servicio que se basa exclusivamente en la verificación por punto de comprobación para la transparencia de fallo, tiene que proporcionar, como parte de ese servicio, medios para que los clientes detecten que se ha producido un rearranque desde un punto de comprobación, y sepan que la información relativa al servicio que ellos tienen puede estar anticuada. Si es necesario asegurar la consistencia entre múltiples objetos computacionales, debe especificarse la transparencia de transacción.

### 8.9.3 Transparencia de ubicación

La transparencia de ubicación oculta, con respecto a un objeto computacional, las ubicaciones en el espacio en las que residen los objetos computacionales con los que aquél interactúa. Esto implica que se pueda identificar las interfaces y ganar acceso a las mismas sin especificar su ubicación en el espacio.

### 8.9.4 Transparencia de migración

La transparencia de migración oculta, con respecto a un objeto computacional, el hecho de que se ha desplazado. Está soportada por la función de migración.

La transparencia de migración puede combinarse con la transparencia de persistencia o con la transparencia de fallo, de modo que un conglomerado no sea reactivado en su lugar de origen, sino que la plantilla del conglomerado se transfiera directamente a la nueva ubicación y el conglomerado se reactive en ese lugar.

### 8.9.5 Transparencia de persistencia

La transparencia de persistencia oculta, con respecto a un objeto computacional, la asignación y la anulación de la asignación de recursos a conglomerados o sus plantillas, y permite la compartición de recursos. Está soportada por las funciones de reactivación y desactivación.

Un objeto de ingeniería básico en un conglomerado puede interactuar con otros objetos. Cuando se desactiva el conglomerado, el objeto de ingeniería básico se guarda en una plantilla de conglomerado, con sus actividades congeladas, y no puede interactuar con otros objetos. La transparencia de persistencia oculta la desactivación y reactivación de plantillas de conglomerados, de manera que los objetos de ingeniería básicos parecen estar siempre disponibles para interacción. Esto implica que un objeto tiene su propio tiempo de vida, independiente de su entorno de soporte.

### 8.9.6 Transparencia de reubicación

La transparencia de reubicación oculta, con respecto a un objeto computacional, el hecho de que las ubicaciones de las interfaces a que está vinculado este objeto han cambiado. Esto implica la aptitud para establecer de nuevo la vinculación, si es necesario. La transparencia de reubicación está soportada por la función de reubicación.

### 8.9.7 Transparencia de replicación

La transparencia de replicación utiliza un grupo de objetos computacionales para soportar una interfaz computacional única. Está soportada por la función de replicación y la función de reubicación.

### 8.9.8 Transparencia de transacción

La coordinación de transacciones implica la calendarización, supervisión y recuperación de las acciones de interés dentro de esas transacciones. Para obtener este control se requieren interacciones entre los objetos computacionales que intervienen en la ejecución de las acciones de interés y los objetos computacionales que realizan la función de transacción. Generalmente no es posible coordinar las acciones de interés simplemente configurando objetos de ingeniería en los canales para interceptar las acciones de interés. En particular, las acciones internas como el comienzo y la terminación de una transacción no pueden detectarse supervisando las interacciones entre objetos computacionales.

Por consiguiente, la supervisión y control de transacciones requerirá por lo general interacciones computacionalmente visibles entre objetos computacionales que representan los objetos de información o esquemas dinámicos (es decir, objetos de aplicación) por una parte, y objetos computacionales que proporcionan la función de transacción por otra parte. No obstante, el procesamiento de transacciones es muy complejo y no es deseable complicar la especificación de la funcionalidad de la aplicación por la adición de las complejas interacciones necesarias para el control de las transacciones.

La transparencia de transacción la proporciona un proceso automático que refina una especificación computacional que no tiene control de transacciones para obtener una especificación computacional que sí tiene control de transacciones.

La naturaleza del refinamiento y el grado de intervención del especificador en el proceso de refinamiento dependerán del mecanismo particular de transacción que se utilice. En los casos típicos, será necesario que las interfaces computacionales adicionales se vinculen a los objetos computacionales que intervienen en la función de transacción y con interacciones en estas interfaces computacionales para coordinar la calendarización, supervisión y recuperación de las acciones de interés para la función de transacción. Habrá que ampliar el comportamiento del objeto computacional para añadir acciones de recuperación, y es posible que haya que ampliar o reemplazar las interfaces que dan acceso a su funcionalidad normal.

## 9 Evaluación de la conformidad

### 9.1 Evaluación de la conformidad y proceso de desarrollo

El desarrollo de un producto va de la realización inicial de uno o más de los requisitos de un sistema ODP hasta la provisión final de un ejemplo de un sistema ODP que cumple esos requisitos. El proceso de desarrollo puede comprender la producción de un número de especificaciones. Una especificación puede servir para la elaboración de un número de especificaciones subsiguientes, utilizando uno o varios tipos de pasos ("transformaciones"), que incluyen:

- traducción; y
- refinamiento.

Las especificaciones se expresan en algún lenguaje natural o formal. La traducción produce una especificación con el mismo significado (generalmente en un lenguaje diferente). El refinamiento, en cambio, produce una especificación con nuevos detalles que sirven para definir el producto de una manera más precisa. Si cada especificación se caracteriza por el conjunto de productos potenciales que podría especificar, la traducción deja el conjunto sin modificación, pero el refinamiento produce un subconjunto.

Tras la instalación y la aceptación para el servicio, un ejemplar (dícese también una instancia) del sistema ODP entra en una fase de uso operacional, durante la cual debe satisfacer las necesidades expresadas en el documento que especifica sus requisitos. La aptitud del producto para satisfacer estas necesidades depende de un número de prácticas seguidas durante cada una de estas fases del desarrollo. Las normas para probar la "calidad" describen conjuntos coherentes de esas prácticas así como la organización de personas, la documentación y las etapas del ciclo de vida a que se aplican. En cada fase se hace una medición de la calidad y se efectúan cambios si se observa que no se alcanza el nivel de calidad deseado. La evaluación de la conformidad proporciona una medida de la calidad, generalmente en la fase en que se realiza la especificación de implementación. Sin embargo, puede utilizarse también en otras fases, o tener implicaciones en ellas.

## **9.2 Evaluación de conformidad: Relaciones relevantes**

Las relaciones entre especificaciones e implementaciones reales de interés para la conformidad se dividen en dos grupos:

- i) relaciones entre especificaciones e implementaciones reales [conformidad (en inglés "conformance")]; y,
- ii) relaciones exclusivamente entre especificaciones [cumplimiento (en inglés "compliance"), refinamiento, consistencia (o coherencia) y validez interna].

La conformidad es una relación entre una especificación y una implementación real, como un ejemplo de un producto. Existe conformidad cuando la implementación satisface los requisitos específicos expresados en la especificación (los requisitos de conformidad). La evaluación de la conformidad es el proceso mediante el cual se determina esta relación.

El cumplimiento es una relación entre dos especificaciones, A y B, que existe cuando la especificación A impone requisitos que son cumplidos por la especificación B (cuando B cumple con A).

La conformidad de una implementación real no siempre se evalúa con respecto a la especificación de "nivel más bajo" (es decir, la implementación) en un proceso de desarrollo de un producto. Es posible que se utilice una especificación de nivel "más alto" (por ejemplo, aquella cuyo refinamiento tuvo por resultado la especificación de la implementación). Tanto en uno como en el otro caso, la correspondencia entre especificaciones sucesivas puede ser importante. Tal correspondencia se determina por las reglas de refinamiento.

Dos especificaciones están unidas por la relación de refinamiento cuando una es un refinamiento de la otra y todos los productos que podrían ser conformes con el refinamiento son también conformes con la especificación de la que se partió para obtener el refinamiento. Con esto se asegura que todas las constricciones que están presentes en la especificación más genérica también lo están en la refinada.

Las especificaciones no siempre están relacionadas por el hecho de haber sido derivadas del mismo conjunto de requisitos. Algunas veces han sido desarrolladas desde dos o más puntos de vista distintos (por ejemplo, especificaciones de lenguajes de punto de vista distintos). En este caso la consistencia entre las especificaciones puede ofrecer interés – es importante que los requisitos de una especificación no estén en contradicción con los de otra. La consistencia es una relación entre dos especificaciones, que existe cuando es posible que haya por lo menos un ejemplo de un producto que pueda ser conforme con ambas especificaciones.

Una especificación es válida cuando no hay contradicción entre sus propiedades y las propiedades implícitas que debe tener la especificación (por ejemplo, cabe esperar que una especificación de protocolo esté exenta de atascos), y cuando hay por lo menos un ejemplo de un producto que podría ser conforme con ella (es decir, que no sea contradictoria consigo misma).

La evaluación de la conformidad es la determinación de la existencia o inexistencia de estas relaciones, sea mediante pruebas (conformidad) o por comprobación de la especificación (cumplimiento, verificación de refinamiento, comprobación de la consistencia, comprobación de la consistencia interna).

## **9.3 Puntos de conformidad y conceptos conexos**

Cuando la conformidad de una realización de una especificación ODP se evalúa mediante pruebas de conformidad, su comportamiento se evalúa (aplicándole estímulos y supervisando todo evento resultante) en puntos (de interacción) específicos. Los puntos utilizados se denominan "puntos de conformidad" y suelen elegirse entre varios de esos puntos cuya ubicación se especifica en la arquitectura del modelo de referencia del procesamiento distribuido abierto (RM-ODP). Estos puntos de conformidad potenciales se denominan puntos de referencia.

Para cumplir con el modelo RM-ODP, las especificaciones de ODP deben contener un enunciado de conformidad que (entre otras cosas) deberá indicar los puntos de referencia que deben utilizarse para las pruebas de conformidad. Se tiene el propósito de que cada punto de conformidad indicado en una especificación sea uno de los puntos de referencia definidos en la arquitectura del modelo.

Además de los conceptos de punto de conformidad, punto de referencia y punto de interacción anteriormente descritos, la norma sobre la metodología y el marco para las pruebas de conformidad OSI [ISO/CEI 9646] define también la noción de un punto de control y observación (PCO, *point of control and observation*) que no es el punto en que, según la definición, debe existir el comportamiento conforme (es decir, un punto de conformidad), sino el punto en que se controla y observa el comportamiento en un punto de conformidad.

Un punto de conformidad se especifica en especificaciones ODP con respecto a las cuales probablemente se reclame la conformidad. Se especifica un punto de control y observación cuando se documenta un determinado medio de prueba de la implementación. La existencia de diferentes métodos puede implicar la identificación de diferentes conjuntos de PCO.

El sistema de procesamiento utilizado para realizar pruebas puede describirse utilizando la nomenclatura del ODP. Debe señalarse que unos PCO de la aplicación probada son esencialmente los puntos de conformidad del sistema empleado para la prueba de esa aplicación. Los puntos de referencia del sistema probador son PCO potenciales, que pueden haber sido especificados con anterioridad para tomarlos en consideración cuando se define un método de prueba.

Los fundamentos y la arquitectura de la Rec. UIT-T X.902 | ISO/CEI 10746-2 y la Rec. UIT-T X.903 | ISO/CEI 10746-3 no definen puntos PCO, o puntos PCO potenciales, separadamente; en lugar de esto pueden utilizarse las nociones de los puntos de conformidad y los puntos de referencia del sistema probador. El emplazamiento de los PCO potenciales en puntos de referencia aseguraría que los propios sistemas probadores fueran sistemas ODP y limitaría la gama de métodos de prueba que pudieran requerir especificación. A medida que se extiende la utilización del procesamiento distribuido abierto, es posible que el término PCO resulte redundante, pues no se distinguirá en la práctica entre PCO potenciales y puntos de referencia.

## 9.4 Especificaciones de conformidad ODP

### 9.4.1 Nivel de abstracción

Los probadores de la conformidad pueden requerir información adicional cuando prueben la implementación de alguna especificación ODP. Esta información se denomina información suplementaria de implementación para pruebas (IXIT) e incluye la información que se requiere para relacionar los conceptos en enunciados de conformidad de implementación (ICS, *implementation conformance statements*) de los ODP, con su realización, en una implementación.

El nivel de abstracción en el que se especifica un punto de conformidad tiene implicaciones en el trabajo requerido para proporcionar una relación de correspondencia IXIT, las constricciones impuestas al proceso de implementación y el grado de independencia con respecto a la implementación:

- Se requiere información suplementaria adicional sobre la especificación para proporcionar una relación de correspondencia de la implementación en niveles de abstracción más altos. Puesto que, tanto la provisión de esta información, como el proceso de implementación implican el enlace de términos de una especificación con equipos utilizados para la implementación, la provisión de la relación de correspondencia IXIT puede requerir un nivel de esfuerzo proporcional al requerido en la totalidad del proceso de implementación. Para los puntos de conformidad definidos utilizando un nivel de abstracción particularmente alto, esto puede representar un costo irrazonable, y redundar en una pérdida de entusiasmo por las pruebas de conformidad.
- Se ha señalado también que todos los PCO deben tener alguna realización explícita y accesible en una implementación real, una característica que no tienen necesariamente otros aspectos de una especificación.

La especificación de puntos de conformidad (para los cuales hay que asignar PCO) representa por tanto una restricción impuesta al proceso de implementación, que será tanto mayor cuanto mayor sea el número de puntos definidos. Cuanto menos relacionada con una implementación esté la especificación de un punto de conformidad (por ejemplo, cuanto más alto sea el nivel de abstracción), tanto mayor será la inconveniencia de proporcionar una representación explícita en la implementación real.

- La utilización de un nivel de abstracción más bajo en la especificación de un punto de conformidad implica que se ha hecho abstracción de un número menor de detalles, de una implementación, por lo que la especificación es menos independiente de la implementación. El mayor detalle implica también que se puede requerir un esfuerzo mayor, en el proceso de evaluación de conformidad, para evaluar cada detalle (aunque se necesitará menos interpretación para proporcionar la información IXIT).

#### 9.4.2 Utilización de múltiples puntos de referencia

La definición de una multiplicidad de puntos de conformidad, por oposición a uno solo, en una especificación, puede requerirse para otros fines que no sean, simplemente, aumentar el número de los aspectos prescriptivos de una especificación.

La especificación de un objeto puede definir varios tipos diferentes de interfaz con su entorno. Por ejemplo, un objeto puede tener interfaces con personas, con otros objetos, con mecanismos de comunicación o con mecanismos de almacenamiento. Cada una de estas interfaces puede caracterizarse por puntos de referencia de diferentes clases (clases perceptual, programático, de interfuncionamiento, y de intercambio), cada uno de los cuales tiene diferentes consecuencias en las pruebas de conformidad. Esto implica que será necesario identificar diferentes puntos de conformidad cuando estén presentes interfaces de estos tipos diferentes.

Incluso cuando se considera un solo tipo de interfaz, se puede tener la intención de que la implementación de un objeto sea accesible en puntos física y lógicamente separados. Asociando puntos de conformidad separados, con cada uno de ellos, se reconocen estos detalles de la implementación, y de este modo, se baja el nivel de abstracción en el que están especificados los puntos de conformidad (porque se prescriben más detalles de una implementación), lo que tiene las consecuencias anteriormente indicadas en 9.4.1. Los puntos de conformidad separados hacen también que la "intención" de tener puntos de acceso separados sea un detalle prescriptivo, más bien que descriptivo (con lo que se asegura la realización de la intención).

Cuando se especifica un objeto en términos de un número de componentes cada uno de los cuales tiene diferentes interfaces con cada uno de los demás, no siempre está claro si el detalle interno existe para prescribir, por implicación, el comportamiento externo del objeto, o si se tiene la intención de que sea prescriptivo (disponiendo el medio de construir el objeto). Sin embargo, cuando se considera que los componentes son un detalle prescriptivo, es necesario definir puntos de conformidad adicionales, por ejemplo, en cada una de las interfaces de los componentes.

### 9.5 Implicaciones de la conformidad de los lenguajes de punto de vista

La utilización de lenguajes de punto de vista ODP en especificaciones ODP tiene varias implicaciones:

- se espera que los probadores de la conformidad evalúen el efecto de las especificaciones en términos de la especificación de ingeniería de una implementación sometida a prueba, lo que, junto con otros factores, requiere que un cliente laboratorio de pruebas suministre información suplementaria para las pruebas, incluyendo:
  - i) IXIT+ICS que relacionen la implementación de los conceptos y estructuras de la especificación de empresa de una implementación con la implementación de su especificación de ingeniería;
  - ii) IXIT+ICS que relacionen la implementación de los conceptos y estructuras de la especificación de información de una implementación con la implementación de su especificación de ingeniería;
  - iii) IXIT+ICS que relacionen la implementación de los conceptos y estructuras de la especificación computacional de una implementación con la implementación de su especificación de ingeniería;
  - iv) IXIT+ICS que relacionen la implementación de los conceptos y estructuras de la especificación de ingeniería de una implementación con la implementación de las opciones elegidas en su especificación de tecnología (esto debe proporcionarse como parte de la especificación de tecnología);
- cada lenguaje de punto de vista permite la especificación separada de diferentes tipos de requisitos, de modo que, por ejemplo, se puedan tratar separadamente los objetivos comerciales, el diseño de sistemas y la utilización de tecnología; esto implica que es posible el desarrollo paralelo de especificaciones separadas;
- para comprobar el desarrollo paralelo de especificaciones hay que evaluar la consistencia de las especificaciones unas con respecto a las otras, cada cierto tiempo, durante un proceso de desarrollo.

Dado que las especificaciones proporcionadas desde distintos puntos de vista pueden ser independientes y, por esa razón, pueden no estar relacionadas entre sí por un refinamiento, la comprobación del refinamiento no proporciona una comprobación completa de la progresión paralela de las diversas especificaciones durante su desarrollo. No obstante, es evidente la conveniencia de efectuar algunas de esas comprobaciones. Este requisito se cumple por la comprobación de la consistencia (véase 8.7). Los objetos descritos por especificaciones de punto de vista pertenecen a menudo al mismo universo de racionalidad: el objeto información en una especificación de información puede aparecer como un argumento de una función en la interfaz con un objeto computacional, por ejemplo; los puntos de conformidad de la especificación de empresa pueden reaparecer como puntos de conformidad en las especificaciones computacional y de ingeniería, y así sucesivamente.

## 9.6 Actividades de evaluación de la conformidad

Durante el proceso de evaluación de la conformidad pueden realizarse, entre otras, las siguientes actividades:

- comprobación del refinamiento, entre especificaciones;
- comprobación de la validez interna de una especificación individual;
- comprobación de la consistencia de varias especificaciones;
- prueba de una realización o animación, entre una implementación real y una especificación.

## 10 Gestión de los sistemas ODP

Un sistema ODP está constituido por un número de aplicaciones ODP junto con sus servicios de soporte. Los servicios de soporte incluyen servicios tales como procesamiento, almacenamiento de ficheros, acceso de usuario y comunicaciones, proporcionados por un sistema operativo tradicional en un sistema centralizado, pero pueden incluir también servicios necesarios para la distribución del sistema, como la gestión de directorios y nombres, comercio (importación/exportación de información), y los servicios de soporte de transparencias de distribución. Todos estos servicios deben ser gestionados, junto con las aplicaciones ODP: la naturaleza de las funciones de gestión necesarias para esto dependerá de los servicios o aplicaciones de que se trate.

La gestión propiamente dicha de un sistema ODP puede efectuarla una o más aplicaciones de gestión interactuando con los servicios de sistema y aplicaciones ODP a través de interfaces que ofrecen un nivel de granularidad de gestión determinado por la aplicación de gestión en cuestión. Cuando sea necesario, la aplicación de gestión interactuará con las funciones ODP a través de las funciones de gestión definidas para ellas.

Las aplicaciones de gestión en un sistema ODP reflejarán interrelaciones entre funciones de gestión; por ejemplo, el hecho de que sea necesario efectuar una reconfiguración para mantener la calidad de servicio demuestra que las funciones de gestión están interrelacionadas. Entre las aplicaciones de gestión se incluye normalmente la de contabilidad, lo que refleja el hecho de que la mayor parte de los servicios necesitan una contabilidad interna para fines de gestión, aunque los usuarios no sean facturados por ese servicio.

Las funciones de seguridad no se incluyen en el conjunto de funciones de gestión, aunque se espera que las interacciones de gestión estén sometidas al control de políticas; por ejemplo, la autenticación y el control de acceso generalmente son esenciales para impedir que usuarios o gestores no autorizados realicen acciones de gestión sobre componentes de un servicio o de una aplicación. Por tanto, todos los servicios en un entorno ODP deben poder utilizar un servicio de seguridad, de la misma manera que deben poder utilizar un servicio de comunicaciones; no obstante, la gestión de seguridad no es un servicio de gestión genérico necesario para la gestión de otros servicios.

### 10.1 Dominios de gestión

Para hacer frente a la complejidad de la gestión y a las cuestiones de escala, especialmente en los grandes sistemas ODP, es necesario proporcionar un marco común para la partición de la gestión global. Dentro del entorno ODP hay una multiplicidad de visiones de la gestión y de demarcaciones de responsabilidad coexistentes, cada una de las cuales se basa en criterios de estructuración diferentes.

Los dominios de gestión proporcionan un medio flexible y pragmático de especificar demarcaciones de responsabilidad y autoridad que reflejen estas diferentes visiones. Un dominio identifica un conjunto de objetos, cada uno de los cuales está unido, por una relación característica, a un objeto controlador de dominio. Los objetos miembros pueden ser recursos, estaciones de trabajo, módems, procesos, etc., lo que dependerá de la finalidad para la cual se ha definido un determinado dominio de gestión. El objeto controlador tiene un atributo que identifica los objetos miembros. Se dice que un objeto es un objeto miembro si su identidad es conocida por el objeto controlador del dominio de gestión.

Los dominios de gestión permiten controlar, con una política común, un conjunto de objetos gestionados, proporcionando una base para hacer frente a la complejidad de los sistemas ODP de gran escala. Simplifican la actividad de gestión porque la política y la calidad de miembro del conjunto pueden modificarse mediante interacciones con un solo objeto – el objeto controlador de dominio de gestión – sin tener que obligar a los gestores a interactuar individualmente con la multiplicidad de objetos gestionados en el entorno.

Los dominios de gestión no encapsulan los objetos miembros – los objetos externos pueden interactuar directamente con un objeto en un dominio. Los dominios de gestión son persistentes, incluso si, en algunos momentos, no contienen ningún objeto, ya que tiene que ser posible crear un dominio de gestión vacío y posteriormente introducirle objetos.

## 10.2 Política de gestión

En términos de empresa, la relación característica de un dominio comprende la política asociada con ese dominio. Así, el dominio de gestión proporciona el medio para especificar una política de gestión para un grupo de objetos gestionados, sin que tenga que hacerse esto para cada objeto individual. El objetivo de gestión global y las constricciones externas impuestas por leyes (por ejemplo, ley de protección de datos), reglamentos, o políticas de nivel superior son dos aspectos de la política para un dominio de gestión. Estos ejemplos muestran que puede ser difícil especificar formalmente algunas políticas.

Las constricciones internas imponen restricciones a las operaciones que pueden realizarse sobre objetos en un dominio de gestión. Éstas pueden expresarse de modo declarativo, imponiendo obligaciones a miembros potenciales del dominio.

Un aspecto importante de la política de gestión es la especificación de las operaciones que los gestores pueden realizar sobre los objetos por ellos gestionados. Una regla de acceso es una relación de autoridad que especifica el conjunto de interacciones permitidas entre un dominio de gestores y un dominio de objetos gestionados. Por ejemplo, todos los miembros del dominio "ProgramadoresDeSistema" están autorizados para poner en marcha y detener los objetos en el dominio "ServiciosDeDepartamento". Las acciones permitidas pueden ser un subconjunto de las interacciones de gestión definidas por las interfaces con los objetos en el dominio.

## 10.3 Modelado de estructuras de gestión

Las relaciones de dominio pueden utilizarse para modelar estructuras de gestión. Dos dominios de gestión son por definición disjuntos si ninguno de los objetos pertenece a ambos dominios. Dos dominios se superponen si existen objetos que pertenecen a ambos dominios. Un ejemplo de esto es el dominio compartido de una central de cabecera que interconecta dos redes a través de los centros de gestión de cada red. Esto puede realizarse referenciado el objeto desde ambos dominios de gestión.

Puede producirse una superposición implícita de dos dominios de gestión que contengan objetos gestionados de tipos diferentes pero que hagan referencia a una misma entidad del mundo real. Un ejemplo de esto es el dominio de calendarización y mantenimiento, donde, la puesta fuera de servicio de una estación de trabajo en el dominio de mantenimiento la hace indisponible en el dominio de calendarización. Una superposición implícita se produce probablemente cuando las funciones de gestión están repartidas entre diferentes dominios de gestión.

## 11 Utilización de normas en sistemas ODP

Esta cláusula tiene por finalidad ilustrar, mediante un sencillo ejemplo, la utilización de normas que no son específicamente *normas ODP*, en un sistema ODP, y donde tal utilización está definida en una especificación de un sistema ODP. Para ello se describe, en grandes líneas, el contenido de cada especificación de punto de vista de un ejemplo de sistema, y para cada especificación de punto de vista se examinan los tipos de norma adecuados para cumplir los requisitos en ella identificados.

El ejemplo de sistema se muestra en la figura 16, que representa un sistema servidor con un operador local, conectado por un medio de telecomunicación a un sistema cliente constituido por una estación de trabajo que tiene dos operadores locales. La descripción del sistema considera una aplicación de estación de trabajo que proporciona, a sus operadores locales, acceso a una aplicación proveedora de imágenes en el sistema servidor. Se supone que tanto el sistema servidor como la estación de trabajo proporcionan otros servicios, y que las organizaciones a que respectivamente pertenecen son distintas, es decir, están sometidas a políticas administrativas distintas.

Las descripciones de punto de vista indican requisitos para especificaciones comunes. Estas especificaciones comunes corresponderán, en general, a normas en niveles diferentes.

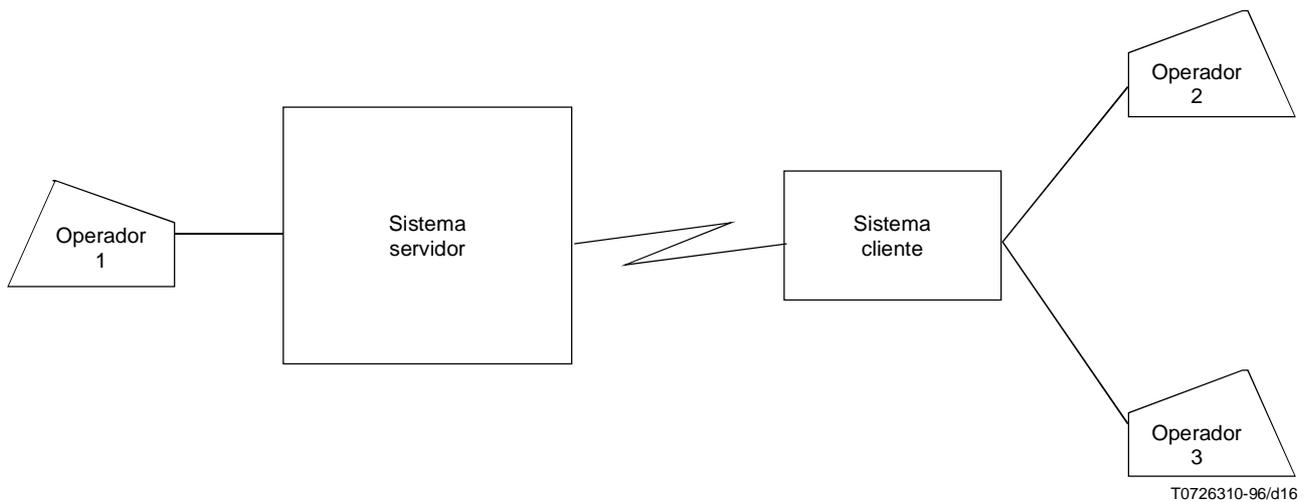
### 11.1 Punto de vista de la empresa

#### 11.1.1 Especificación de empresa

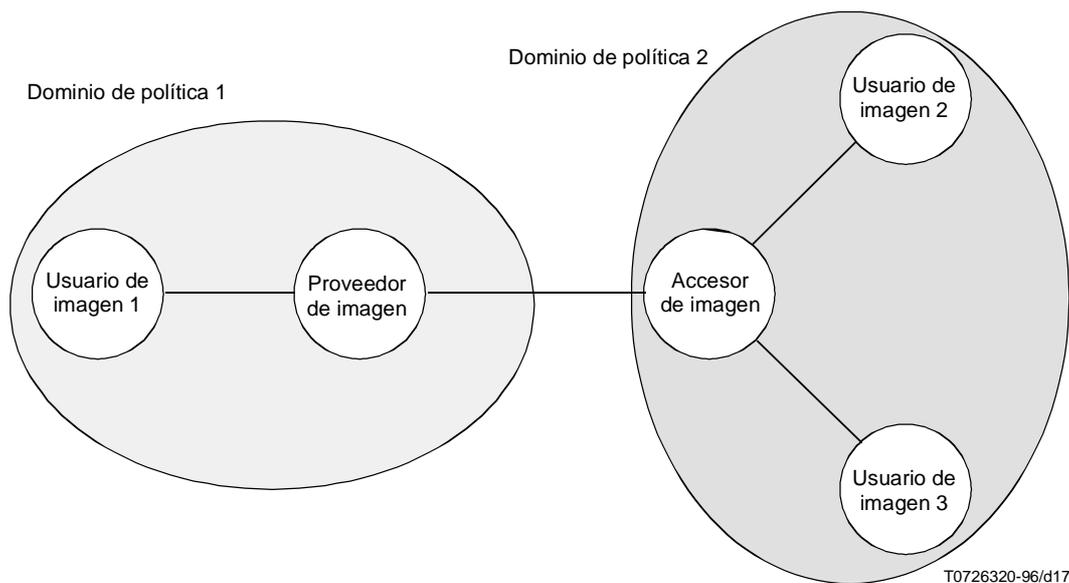
La figura 17 ilustra el modelo de objeto para la especificación de empresa del sistema representado en la figura 16, donde:

- Usuario de imagen n es un objeto de empresa que representa una persona y está desempeñando el rol "usuario de imagen n".
- Proveedor de imagen es un objeto de empresa que representa un sistema IT (tecnología de la información) y está desempeñando el rol "proveedor de imagen".

- Accesor de imagen es un objeto de empresa que representa un sistema IT y está desempeñando el rol "accesor de imagen".



**Figura 16 – Ejemplo de configuración de sistema**



**Figura 17 – Punto de vista de la empresa**

El objetivo de la empresa representado por la configuración de objetos de empresa rige la naturaleza del comportamiento asociado con los roles desempeñados por esos objetos. Por ejemplo, para alcanzar el objetivo podría requerirse solamente que hubiera una visualización de imagen fija para los usuarios de imagen 2 y 3, o que el accesor de imagen soportara la manipulación de la información de imagen, la adición de información de imagen y la devolución de información, en reacción, al proveedor de imagen.

Así, la especificación de empresa define el objetivo de la configuración de objetos de empresa, y por tanto:

- las necesidades de información de imagen de los usuarios de imagen 1, 2 y 3;
- las interacciones del usuario de imagen 1 con el proveedor de imagen;

- las interacciones del usuario de imagen 2 y del usuario de imagen 3 con el accesor de imagen;
- las interacciones entre el accesor de imagen y el proveedor de imagen;
- las políticas (incluida la de seguridad) que rigen las interacciones entre objetos de empresa;
- los requisitos de calidad de servicio de los usuarios de imagen 1, 2 y 3.

### 11.1.2 Aplicación de normas

En la descripción de empresa podría elaborarse especificaciones comunes para un aspecto específico de una función de industria o de negocio, como una *política de seguridad*, que debe ofrecerse en términos de controles de acceso y de fuerza de protección. Tales normas se relacionarían con las áreas de negocio y de aplicación en cuestión.

## 11.2 Punto de vista de la información

### 11.2.1 Especificación de información

El sistema es visible desde el punto de vista de la información en términos de:

- las clases de objeto de información que intervienen en la aplicación;
- las actividades de información (cambios de estado de objetos de información) que constituyen la aplicación;
- las restricciones impuestas a los cambios de estado que pueden producirse en los objetos de información.

La especificación de información incluye:

- Especificación de las clases de objeto información propiamente dichas, por ejemplo:
  - plantillas de objeto de información de imagen que determinan las clases de objeto de información de imagen que están disponibles;
  - plantillas de objeto de visualización de imagen que determinan las clases de objeto de visualización de imagen que están disponibles. Una plantilla de objeto de visualización de imagen define una composición de objetos de información de imagen;
  - una plantilla de objeto de información de petición, donde un objeto de información de petición comprende la información necesaria para solicitar una visualización, incluida información de control (por ejemplo, para seguridad);
  - un objeto de *información de control de acceso* que comprende la información necesaria para validar un objeto de información de petición.
- Constricciones impuestas a las configuraciones de objetos de información, por ejemplo:
  - el conjunto de clases de plantillas de objetos de información de imagen para el accesor de imagen es el mismo que el conjunto para el proveedor de imagen;
  - el conjunto de clases de plantillas de objetos de visualización de imagen para el usuario de imagen 1 es un subconjunto del conjunto para el proveedor de imagen;
  - los conjuntos de clases de plantillas de objetos de visualización de imagen para los usuarios de imagen 2 y 3 son subconjuntos del conjunto para el accesor de imagen.

Para el sistema en cuestión, un ejemplo (simplificado) de una actividad de información podría ser:

- la creación de un objeto de información de petición con estado invalidado;
- la interacción de un objeto de información de petición y de un objeto de información de control de acceso y cambio de estado, a validado, de un objeto de información de petición;
- interacción de un objeto de información de petición con objetos de información de imagen y creación de objetos de visualización de imagen.

### 11.2.2 Aplicación de normas

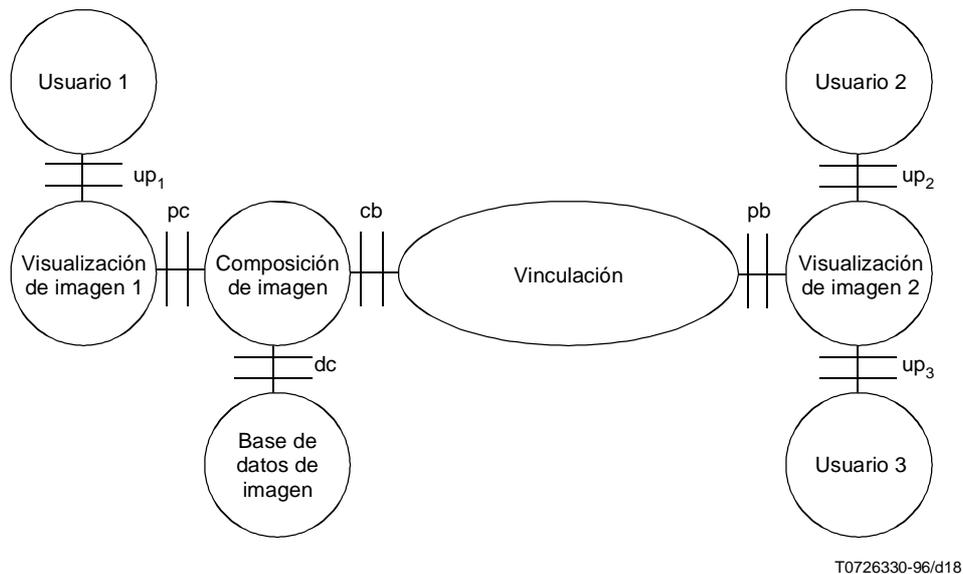
En la descripción de información podrían elaborarse especificaciones comunes para objetos de información. Esas normas se relacionarían con las áreas de negocio y de aplicación en cuestión.

## 11.3 Punto de vista computacional

### 11.3.1 Especificación computacional

La figura 18 ilustra el modelo de objeto para la especificación computacional del sistema representado en la figura 16. El modelo de objeto comprende una configuración de:

- los objetos computacionales usuario 1, 2 y 3, visualización de imagen 1 y 2, composición de imagen y base de datos de imagen;
- vinculaciones primitivas, para las cuales no existe un enunciado explícito de contratos de entorno, entre interfaces de:
  - usuario 1 y visualización de imagen 1 ( $up_1$ );
  - usuarios 2 y 3, y visualización de imagen 2 ( $up_2$  y  $up_3$ );
  - visualización de imagen 1 y composición de imagen ( $pc$ );
  - base de datos de imagen y composición de imagen ( $dc$ );
- una vinculación compuesta en que interviene un objeto de vinculación (Vinculación) entre las interfaces de visualización de imagen 2 y composición de imagen, con un enunciado de los contratos de entorno;
- las actividades especificadas para la configuración de objetos computacionales que realizan los requisitos especificados en las especificaciones de empresa y de información. Las interfaces son puntos de referencia para conformidad en términos del ODP.



**Figura 18 – Punto de vista computacional**

El sistema es visible desde el punto de vista computacional en términos de:

- especificaciones de los objetos computacionales en términos de especificaciones abstractas de las operaciones y comportamiento que soportan en sus interfaces;
- una especificación de la plantilla del objeto de vinculación, que incluye:
  - una especificación abstracta de las operaciones de interfaz que intervienen;
  - una especificación de contratos de entorno en consonancia con los requisitos calidad de servicio de la empresa;

- una especificación de los tipos de datos abstractos que corresponden a los objetos de información identificados a partir del punto de vista de la información;
- especificaciones de las actividades que pueden realizarse para soportar la aplicación.

### 11.3.2 Aplicación de normas

En la descripción computacional podrían elaborarse especificaciones comunes para las interfaces identificadas en la figura 18, especificaciones que se definirían en términos de operaciones abstractas y tipos de datos abstractos (correspondientes a los objetos de información en la descripción de información). Las normas para las operaciones abstractas podrían ser comunes a varias áreas de aplicación; las normas para los tipos de datos abstractos se definiría para el área de aplicación en cuestión. Por ejemplo, una especificación que se basa en OMG/CORBA y se describe en OMG/CORBA IDL es una descripción computacional.

## 11.4 Punto de vista de la ingeniería

### 11.4.1 Especificación de ingeniería

La figura 19 ilustra (en parte) el modelo de referencia para la especificación de ingeniería del sistema representado en la figura 16. En este modelo basado en objetos:

- el objeto de ingeniería básico usuario 1 representa al operador 1;
- los objetos de ingeniería básicos de visualización de imagen, composición de imagen y base de datos de imagen corresponden a los objetos computacionales visualización de imagen, composición de imagen y base de datos de imagen;
- los objetos de stub, vinculador y protocolo constituyen una parte de un canal que corresponde al objeto de vinculación en la descripción computacional;
- el nodo corresponde al sistema servidor en la figura 16 (aunque la figura 19 ilustra solamente una parte de la configuración completa de cápsulas, conglomerados, canales y otros recursos para el nodo).

El sistema es visible desde el punto de vista de la ingeniería en términos de:

- especificaciones del comportamiento de los objetos de ingeniería en el canal correspondiente al objeto de vinculación en el punto de vista computacional, incluyendo:
  - especificación de los protocolos en el *punto de referencia de interfuncionamiento* entre los objetos de protocolo (o entre el objeto de protocolo y el interceptador);
  - especificación de la representación concreta de los tipos de datos abstractos identificados en la descripción computacional;
  - requisitos de calidad de servicio;
- para cada punto de referencia, una especificación de la sintaxis en términos de la cual se expresa el comportamiento en ese punto de referencia;
- constricciones impuestas al comportamiento en los puntos de referencia que reflejan las actividades computacionales especificadas.

### 11.4.2 Aplicación de normas

En la descripción de ingeniería podrían elaborarse especificaciones comunes para aplicación en los puntos de referencia identificados en la figura 19:

- para especificaciones de protocolo, y para especificaciones de sintaxis de transferencia abstractas y concretas, que se aplican en el punto de referencia de interfuncionamiento dentro del canal (que corresponde al objeto de vinculación en la descripción computacional), por ejemplo:
  - se puede aplicar perfiles OSI (por ejemplo un perfil A de acceso a base de datos distante (RDA), un perfil T apropiado y perfiles F apropiados) para protocolos y para sintaxis de transferencia abstractas y concretas, para el establecimiento y mantenimiento del canal;
  - hay que definir, para el área de aplicación en cuestión, normas y perfiles F para las sintaxis abstractas y concretas que corresponden a los tipos de datos abstractos en la descripción computacional.

NOTA – Los protocolos y sintaxis concretas en el punto de referencia de interfuncionamiento incluyen los necesarios para establecer y mantener el canal, así como los que corresponden a las operaciones abstractas y a los tipos de datos abstractos especificados para el objeto de vinculación.

- para especificaciones API (incluidas especificaciones de sintaxis abstractas para datos) que se aplican en los puntos de referencia programáticos, por ejemplo:
  - un perfil SQL se puede aplicar en el punto de referencia entre el objeto de composición de imagen y el objeto de base de datos de imagen;
  - un perfil API para el servicio soportado por el perfil RDA se puede aplicar en el punto de referencia entre el objeto de composición de imagen y el objeto de stub;
  - un perfil API para un servicio de presentación de ventanas (*windowing*) se puede aplicar en el punto de referencia entre el objeto de composición de imagen y el objeto de visualización de imagen (podría incluir normas gráficas);
- para especificaciones de interfaz persona-computador (interfaz HCI) que se aplican en el *punto de referencia perceptual* – una norma de interfaz de usuario gráfico (interfaz GUI), por ejemplo.

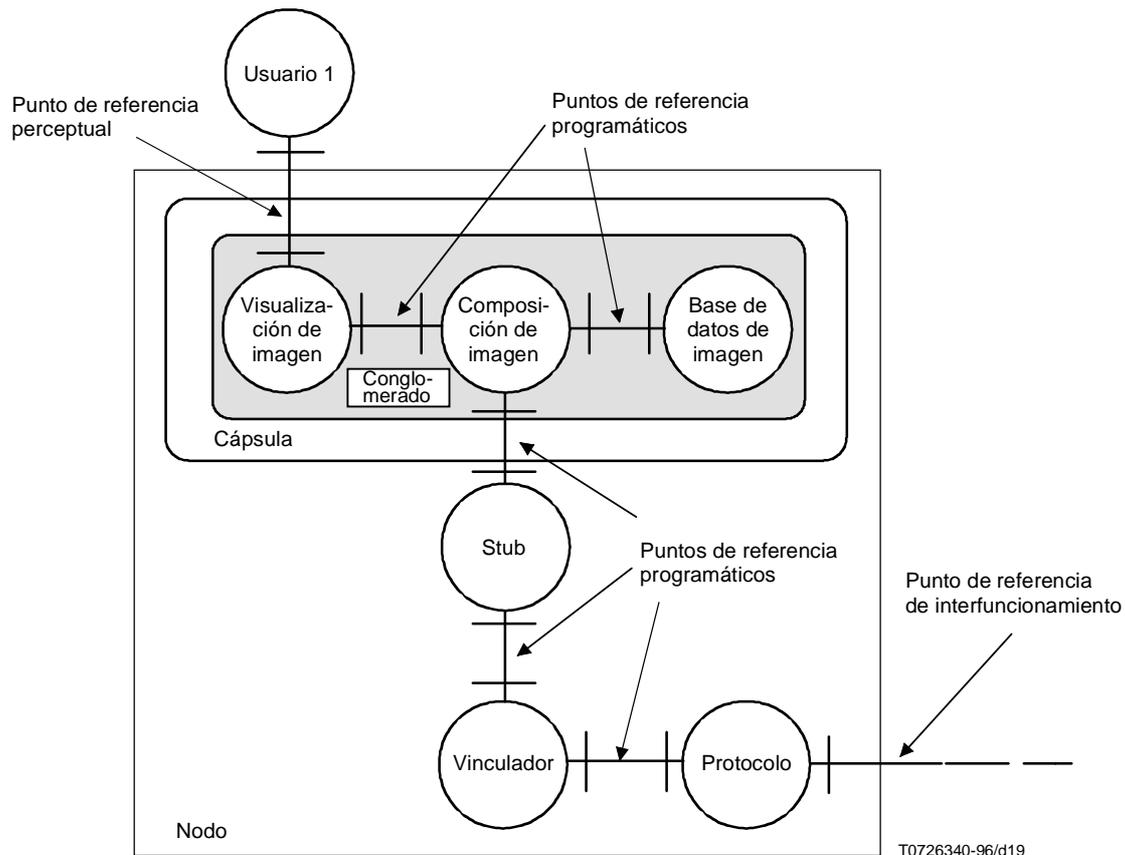


Figura 19 – Punto de vista de la ingeniería (en parte)

## 11.5 Punto de vista de la tecnología

### 11.5.1 Especificación de tecnología

El sistema es visible desde el punto de vista de la tecnología en términos de enunciados (*statements*) del suministrador sobre la conformidad de su sistema. Esto se expresará:

- identificando, para el sistema suministrado, puntos de conformidad que corresponden a puntos de referencia en la descripción de ingeniería;
- indicando la información suplementaria para probar la conformidad del sistema con el comportamiento especificado para esos puntos de conformidad.

Obsérvese que no se requiere que el suministrador haga visibles todos los puntos de referencia como puntos de conformidad.

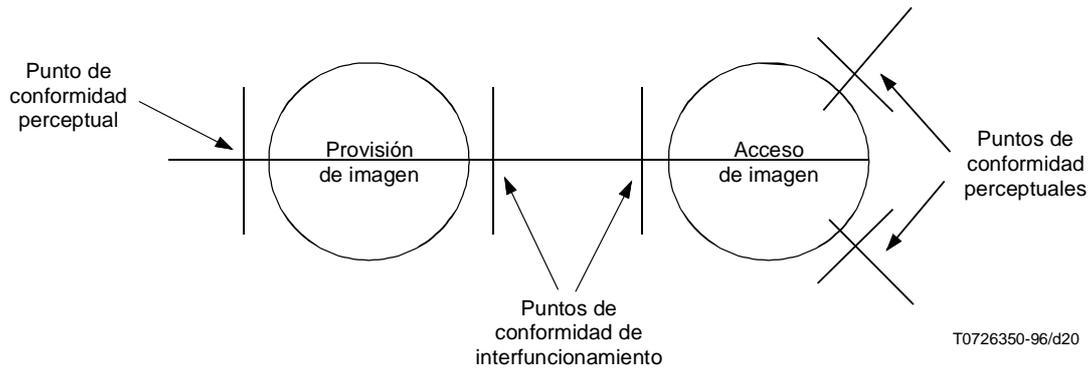


Figura 20 – Punto de vista de la tecnología (en parte)

### 11.5.2 Aplicación de normas

En la descripción de tecnología:

- una implementación de un protocolo OSI específico es un ejemplo de la aplicación de normas OSI;
- los puntos de conformidad identifican los puntos en los que puede observarse el comportamiento del sistema;
- las normas que se aplican en esos puntos de conformidad especifican la sintaxis y el orden de los intercambios con arreglo a los cuales se expresa el comportamiento.

Además de esto, se imponen constricciones al comportamiento en todos los puntos de conformidad al exigir que el comportamiento del sistema sea consistente con las normas especificadas para los otros puntos de referencia en la descripción de la ingeniería.

## 12 Ejemplos de especificaciones ODP

En esta cláusula se presentan varios ejemplos de la utilización, en especificación de sistema, de los conceptos y reglas descritos en la Rec. X.902 UIT-T | ISO/CEI 10746-2 y la Rec. UIT-T X.903 | ISO/CEI 10746-3. Los ejemplos son simplificaciones y descripciones incompletas de sistemas en la vida real, pero sirven para dar una idea de la utilización del marco del ODP, ilustran en un alto nivel la aplicación de los conceptos esenciales de cada lenguaje de punto de vista, así como las relaciones entre las descripciones de punto de vista.

En el ejemplo de 12.1 se utilizan conceptos y reglas para diseñar un **sistema de conferencias multimedia** (MMCS, *multimedia conferencing system*) en los cinco puntos de vista. El MMCS en cuestión permite el interfuncionamiento en tiempo real entre varios usuarios que emplean información multimedia como texto, vídeo y audio.

En el ejemplo de 12.2 se especifica el **intercambio de audio/vídeo entre múltiples partes** en sistemas distribuidos (un componente específico de un MMCS). Se utiliza el concepto de "vinculación de tren" como base para el intercambio de flujos de audio y vídeo de múltiples partes, y se proporciona una especificación empleando los cinco lenguajes de punto de vista ODP. Se presta especial atención a las correspondencias entre las cinco especificaciones de punto de vista, con el fin de asegurar la consistencia entre las especificaciones.

En el ejemplo de 12.3 se sitúan **conceptos de gestión** en el marco del modelo de referencia ODP.

En el ejemplo de 12.4 se da una visión de conjunto de la especificación de una **base de datos distribuida**.

Los siguientes conceptos se ilustran en las diferentes especificaciones de punto de vista de los ejemplos:

**Empresa:**

- comunicación/federación;
- la asociación de roles con objetos de empresa;
- contrato, plantilla y política.

**Información:**

- esquema estático;
- esquema dinámico;
- esquema invariante.

**Computacional:**

- especificación de objeto computacional, incluido contrato de entorno, comportamiento;
- especificación de interfaz de operación y de tren;
- concepto de vinculación;
- referencias de interfaz y reglas de interacción;
- transparencias.

**Ingeniería:**

- establecimiento de canal (objetos de protocolo, vinculador y stub);
- utilización de reglas de estructuración y funciones de ingeniería para especificar la infraestructura acorde con las especificaciones de empresa, de información y computacional.

**Tecnología:**

- opciones para elegir componentes de soporte físico y de soporte lógico acordes con las otras especificaciones de punto de vista;
- identificación de puntos de conformidad implicados por las opciones de tecnología.

## 12.1 Sistema de conferencias multimedios

### 12.1.1 Introducción

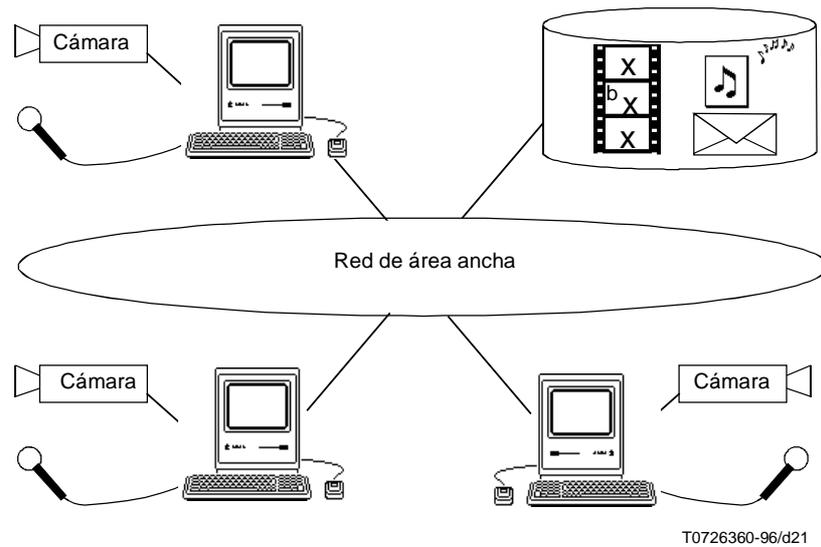
El sistema de conferencias multimedios (MMCS, *multimedia conferencing system*) permite el interfuncionamiento en tiempo real entre varios usuarios que emplean información multimedios como texto, vídeo y audio. El servicio permite a un grupo de personas, que están físicamente distribuidas, trabajar conjuntamente en un documento (multimedios) y comunicarse unas con otras. En el curso de una sesión pueden incorporarse a la conferencia nuevos participantes, y los participantes actuales pueden separarse de la conferencia.

El servicio proporciona asimismo un control de usuario sobre varios atributos de servicio como el tipo o tipos de información deseados, la calidad de un tipo de información, y otros. El MMCS proporciona también un marco para diversas aplicaciones que deben cooperar.

Las aplicaciones como conferencias de vídeo/audio, edición conjunta y correo electrónico deben estar integradas desde la perspectiva de los usuarios. Deberá ser posible ampliar el MMCS para hacer que sea un sistema abierto.

La configuración MMCS en la figura 21 puede constar de estaciones de trabajo para representar vídeo, texto y audio. Existen además cámaras, micrófonos y una base (posiblemente distribuida) de datos multimedios conectados a una red área ancha.

Para la especificación de este ejemplo, los conceptos y reglas pertinentes para la especificación del MMCS se aplicarán para cada punto de vista ODP. No está dentro del ámbito de este ejemplo describir todos los aspectos de un MMCS, sino más bien ilustrar cada lenguaje de punto de vista para especificar una aplicación distribuida abierta o un sistema distribuido abierto. Se utiliza un método de análisis basado en objetos, OMT [Rumbaugh 91], para expresar las especificaciones ODP de empresa y de información.



**Figura 21 – Configuración de un sistema de conferencias multimediales**

### 12.1.2 Especificación de empresa

La especificación de empresa describe los objetivos, políticas y requisitos del MMCS en cuestión. Los requisitos y políticas del servicio MMCS están determinados por las partes que intervienen. Pueden clasificarse de acuerdo con su rol:

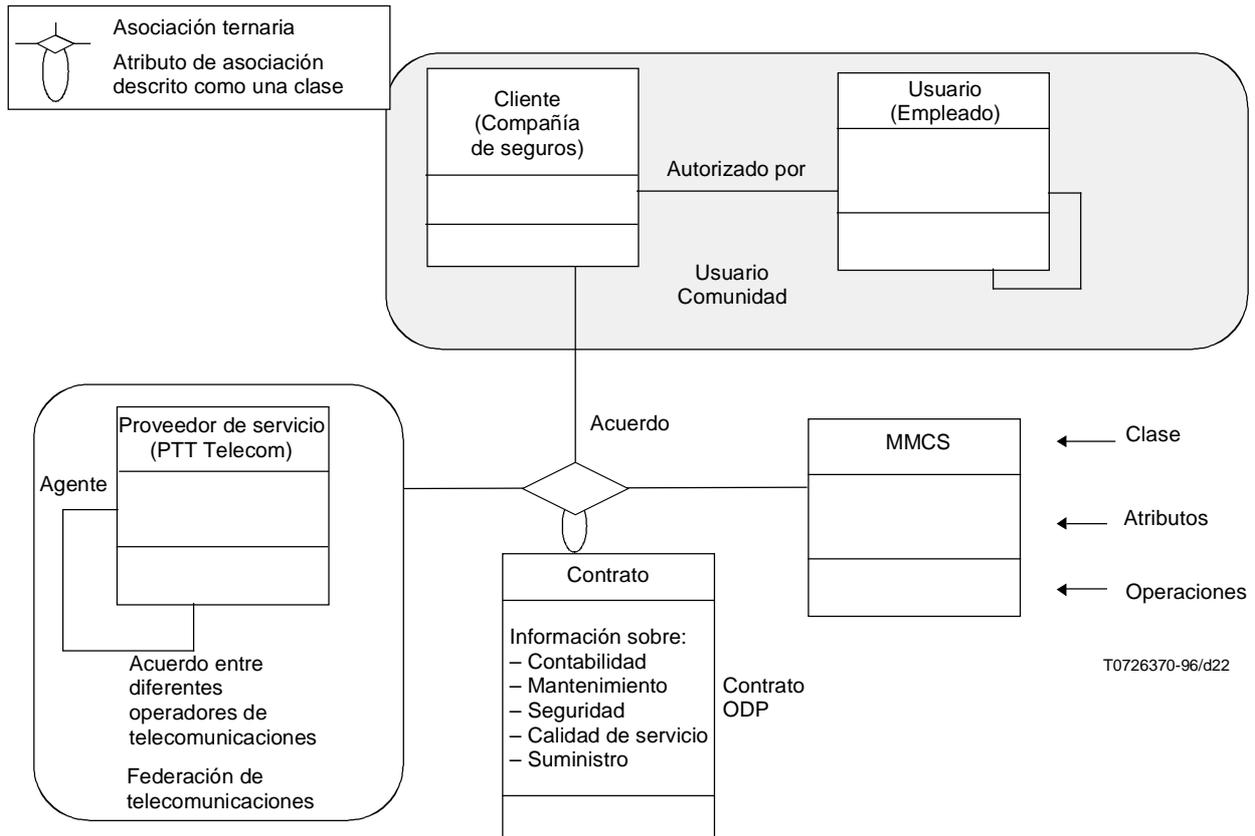
- Usuario: Una persona o aparato que utiliza servicios para satisfacer algunas necesidades de comunicación.
- Cliente o abonado: Una persona u organización que contrata servicios ofrecidos por proveedores de servicio(s).
- Proveedor de servicio(s): Una organización que gestiona comercialmente servicios ofrecidos a clientes en base a un acuerdo contractual.

Para estructurar los requisitos de un determinado servicio de acuerdo con los roles puede necesitarse una especialización de los roles genéricos antes indicados. Por ejemplo, en el caso del MMCS pueden distinguirse los dos tipos siguientes de usuarios: usuario participante en la sesión y usuario director de la sesión.

Surge entonces esta cuestión: "¿qué debe describirse para cada rol que participa en el servicio?" El RM-ODP da indicaciones que abarcan una amplia gama de políticas y reglas de interés para la descripción de servicios distribuidos. Las reglas y políticas genéricas de interés son las siguientes: reglas de utilización de recursos, reglas de dominios, reglas de resolución de conflictos, reglas de organización, reglas de negocios, reglas de transferencia, reglas de seguridad, reglas de calidad de servicio y reglas de gestión.

La figura 22 ilustra una especificación de una empresa sencilla en la que se utilizan conceptos de empresa ODP, expresados en notación gráfica OMT.

Un servicio de conferencias multimediales se proporciona en varios países diferentes. Esto puede interpretarse como varios usuarios que interactúan con su operador telecomunicaciones (proveedor de servicio). Con el fin de expresar el MMCS en términos de ODP, esto puede representarse utilizando los conceptos de comunidad y federación. Puede decirse que los usuarios existen en la comunidad de usuarios y que los operadores de telecomunicaciones pertenecen a una federación. Se puede explorar las relaciones e interacciones entre ambas y, con respecto al MMCS, establecer un acuerdo por medio de un contrato.



**Figura 22 – Especificación de empresa ODP en la notación gráfica OMT**

El usuario, el proveedor de servicio y el MMCS pueden describirse, en términos del ODP, como objetos de empresa con roles asociados. Por ejemplo, el usuario puede ser llevado a participar en acciones de realización (dícese también acciones performativas) con el MMCS a través del cliente, y viceversa. Estas interacciones cambian las denominadas relaciones de obligación, permiso y prohibición entre el usuario y el MMCS. Cada uno de los usuarios del grupo de usuarios interactúa con cada uno de los demás para formar una comunidad ODP porque los usuarios tienen un contrato común de obligación entre roles desempeñados por los objetos de empresa y un conjunto de actividades.

### 12.1.3 Especificación de información

En la especificación de información se especifican la semántica y los requisitos para el procesamiento de la información de servicio.

Para esto se utiliza el concepto de *esquema*. Se define un esquema local para cada rol de usuario, y se especifica un esquema global que representa una información que es válida para todos los roles de usuario y representa información concerniente al servicio. Obsérvese que esta especificación de información presenta un ejemplo para un esquema estático, pero no para un esquema dinámico o invariante.

Sobre la base de la clase MMCS identificada en la especificación de empresa, la especificación de información OMT mostrada en la figura 23 representa un esquema estático para el sistema en un momento en que existe una sesión.

Se describen parámetros de MMCS para la configuración de cliente. Asimismo, se incluye información sobre los límites de la atribución de anchura de banda, una lista de los usuarios de extremo registrados habilitados para iniciar una teleconferencia, una lista de las opciones autorizadas para usuarios de extremo registrados, etc. Esta información de usuario puede considerarse como atributos de un cliente en un modelo de objeto OMT. Un objeto de información de usuario se conecta a un objeto de información de cliente en el sentido de que aquél puede participar en una conferencia solamente si, en la especificación de empresa, el usuario de extremo correspondiente está autorizado para ello por el cliente correspondiente. El usuario director es el usuario que controla la conferencia.

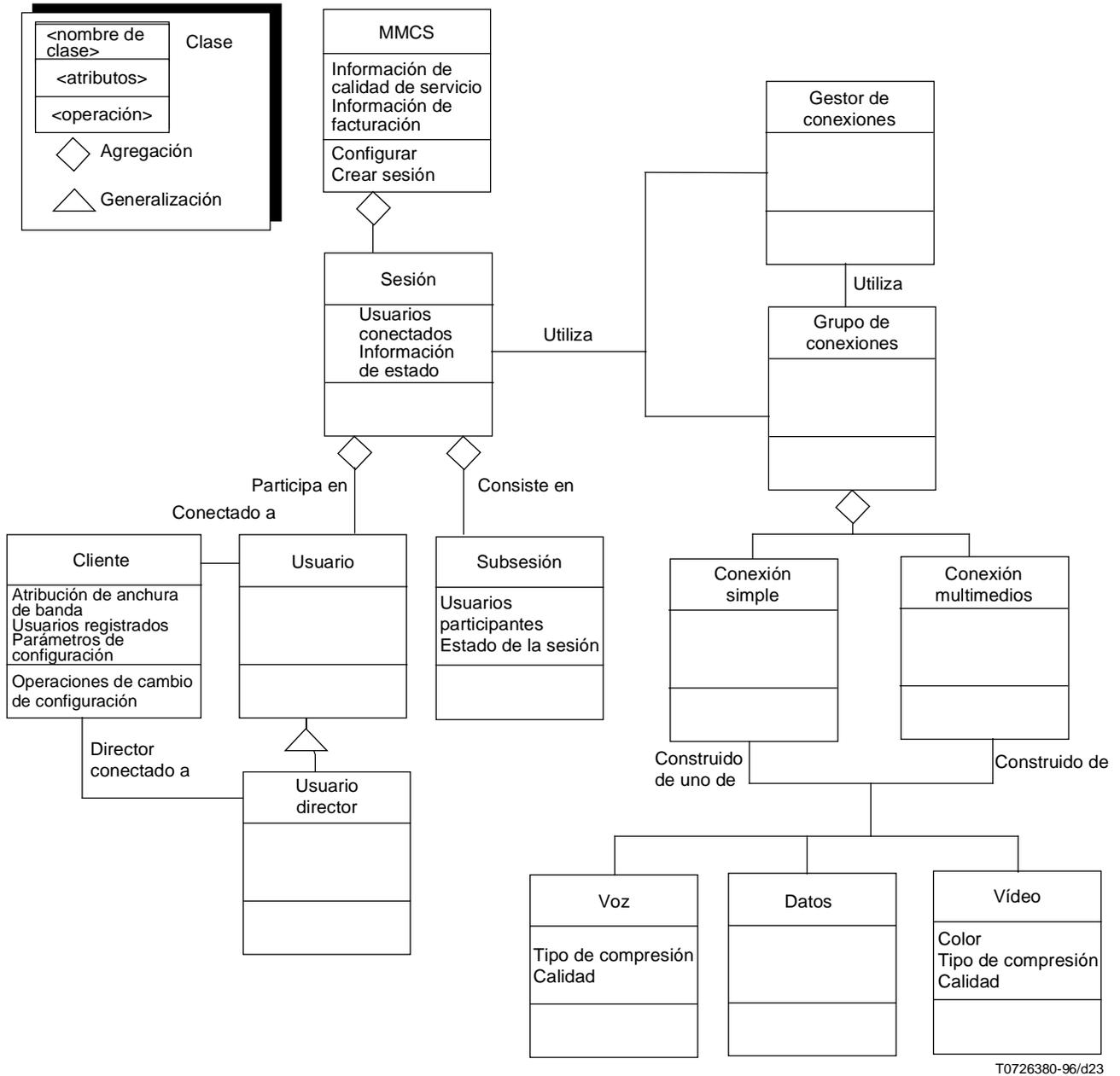
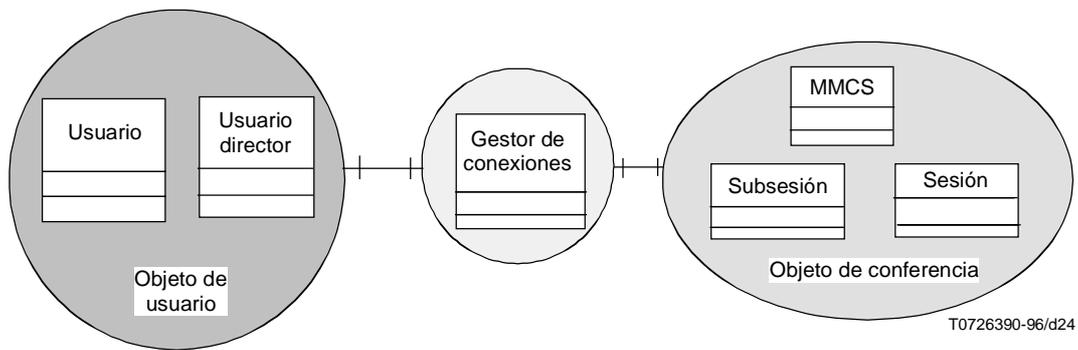


Figura 23 – Especificación de información

12.1.4 Especificación computacional

La especificación de información se elabora de manera que sea consistente con las especificaciones de empresa y de información. La correspondencia entre objetos de información y objetos computacionales no es necesariamente biunívoca. La especificación de información es esencialmente diferente de una especificación computacional; en particular, los objetos computacionales se especifican en términos de interfaces y los objetos de información no.

Las correspondencias entre la especificación de información y la especificación computacional hay que especificarlas en cada caso para que pueda determinarse la consistencia entre las especificaciones. Ésta es una importante tarea que deberá realizar el diseñador del servicio. La figura 24 muestra las correspondencias para el MMCS.



**Figura 24 – Configuración computacional del MMCS**

Se utiliza el análisis OMT para identificar y diseñar los objetos computacionales. Esos objetos computacionales se identifican al reagrupar elementos que están funcionalmente enlazados. El análisis OMT permite también algunas opciones en cuanto a la configuración de ingeniería y el soporte tecnológico.

La decisión sobre la agrupación de clases en objetos computacionales la toma el diseñador del servicio sin atender a los aspectos de distribución:

- los objetos, clases y asociaciones OMT relativos al usuario y al usuario director se tienen en cuenta para el diseño del objeto de usuario computacional;
- los objetos, clases y asociaciones OMT relativos a la conferencia (por ejemplo, MMCS, sesión, subsesión) se tienen en cuenta para el diseño del objeto de conferencia computacional;
- los objetos, clases y asociaciones OMT relativos a la conexión (por ejemplo, gestor de conexiones) se tienen en cuenta para el diseño de las vinculaciones computacionales y serán útiles para la elección de la configuración de ingeniería (por ejemplo, facilidades de multidifusión).

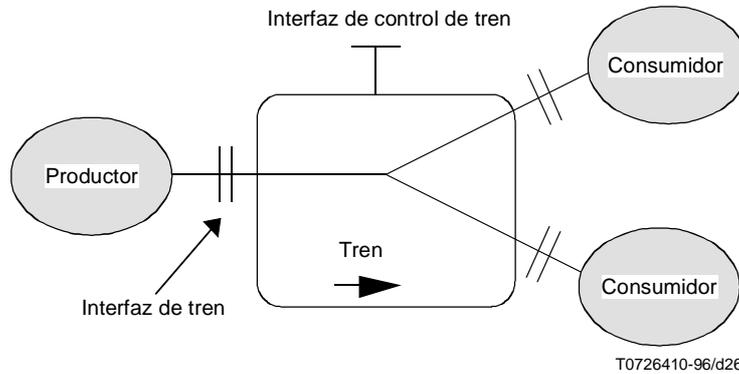
Para cada objeto computacional identificado como resultado de este análisis es necesario definir interfaces para sus interacciones con otros objetos computacionales (por ejemplo, operaciones de conferencia u operaciones para enviar flujos de audio/vídeo). En general, los objetos e interfaces pueden representarse gráficamente como se muestra en la figura 25.



**Figura 25 – Una representación computacional de objetos e interfaces**

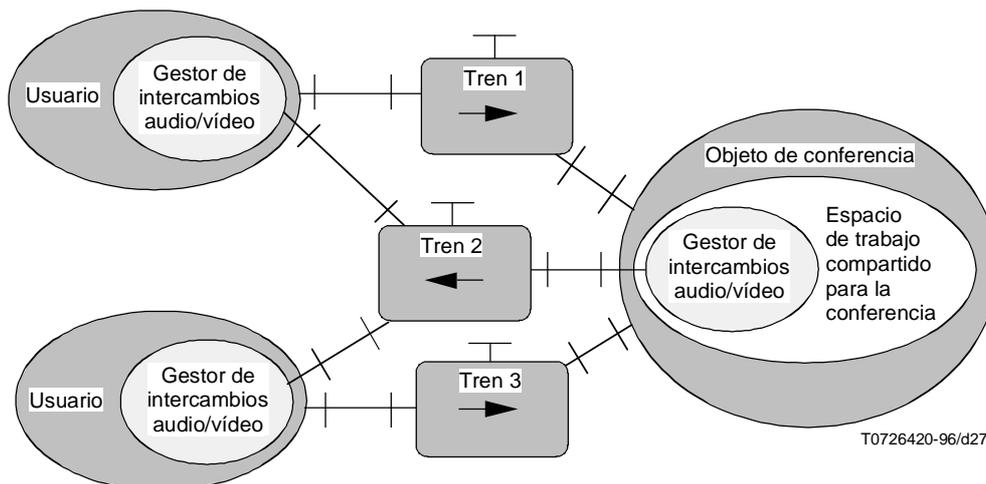
Atendiendo principalmente a las interacciones que son posibles mediante vinculación de objetos, dichas interacciones se establecen por acciones de vinculación explícita que permiten al usuario especificar la vinculación requerida entre los objetos computacionales. Un objeto de vinculación resultante de una vinculación explícita puede soportar un intercambio de operaciones entre objetos computacionales, o un tren si el intercambio concierne a flujos de información continuos.

La figura 26 muestra la notación gráfica utilizada para representar una vinculación explícita, en este caso la vinculación de tren. El objeto de vinculación controla y gestiona las interacciones entre las interfaces de objetos computacionales que abarca. Las operaciones de control se efectúan a través de la interfaz de control de vinculación.



**Figura 26 – Representación de una vinculación explícita (por ejemplo, de un tren)**

La representación computacional de objetos computacionales que intervienen en un intercambio de audio/vídeo se muestra en la figura 27. Se identifican dos objetos principales: el objeto usuario y el objeto conferencia.



**Figura 27 – Configuración MMCS de objetos que intervienen en el intercambio de audio/vídeo**

El objeto de usuario permite la conexión de usuarios de extremo a la conferencia y proporciona a los usuarios de extremo las herramientas necesaria para interactuar entre sí a través del objeto de conferencia y de otros objetos de usuario. El objeto de usuario proporciona a los usuarios de extremo operaciones tales como la incorporación a la conferencia, facilidades de edición, así como herramientas para el intercambio de vídeo y audio.

El objeto de conferencia contiene las funcionalidades requeridas para una conferencia multimedios. Contiene en particular un espacio de trabajo compartido para la conferencia, que se encarga de despachar flujos de audio y vídeo entre objetos de usuario.

Los gestores de intercambio de audio/vídeo tienen a su cargo el envío y la recepción de audio/vídeo durante la conferencia.

Los objetos de tren 1 y tren 3 representan el flujo de audio/vídeo desde los objetos de usuario hasta el espacio de trabajo compartido. El tren 2 representa la multidifusión de flujos de audio/vídeo a todos los objetos de usuario.

A través de las interfaces de control de tren se ejecutan diferentes acciones de control sobre los flujos, como el control dinámico de la calidad de servicio y la sincronización de flujos de audio y vídeo.

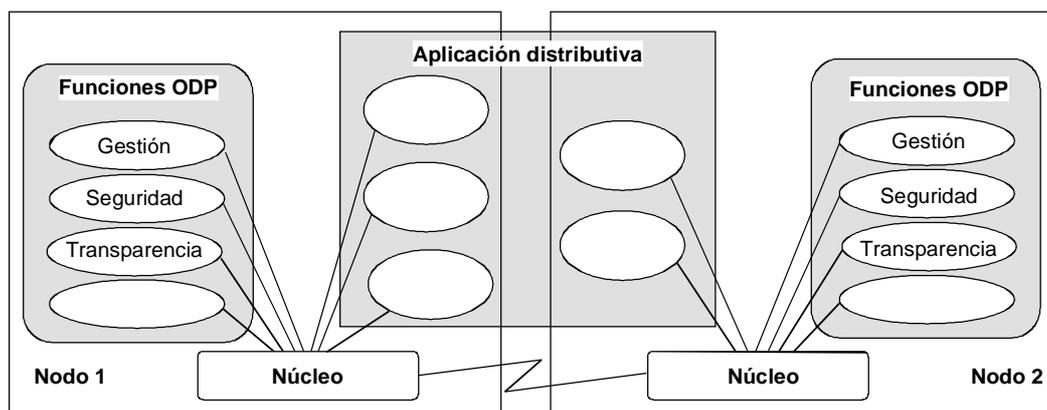
### 12.1.5 Especificación de ingeniería

El lenguaje de ingeniería ODP permite el modelado de la máquina de servicio que soporta la ejecución de la especificación computacional. La figura 28 muestra una arquitectura de ingeniería simplificada. El elemento primordial de esta arquitectura es el núcleo, que controla la utilización de los recursos y permite la comunicación entre diferentes objetos de ingeniería. Algunas funciones, denominadas funciones ODP, comunes a una amplia gama de servicios distribuidos como las funciones de comercio y gestión, están disponibles para aplicaciones distribuidas.

El servicio distribuido se compone de objetos de ingeniería básica que son la representación en la fase de ejecución (pieza de código ejecutable escrito en C++) de una especificación computacional. Una vinculación entre objetos situados en núcleos diferentes se refleja por medio de un canal entre esos objetos. En la figura 28 se representa el correspondiente soporte de ingeniería, destacándose los gestores de audio/vídeo y el tren 2 de la figura 27.

Se establece un canal multipunto (representación de ingeniería del tren) entre los gestores del intercambio de audio/vídeo. Este canal se enlaza con los núcleos que intervienen.

Los objetos de stub proporcionan funciones de adaptación para soportar la transparencia de la distribución (por ejemplo, conversión del formato de datos de una codificación de señal de vídeo a otra diferente).

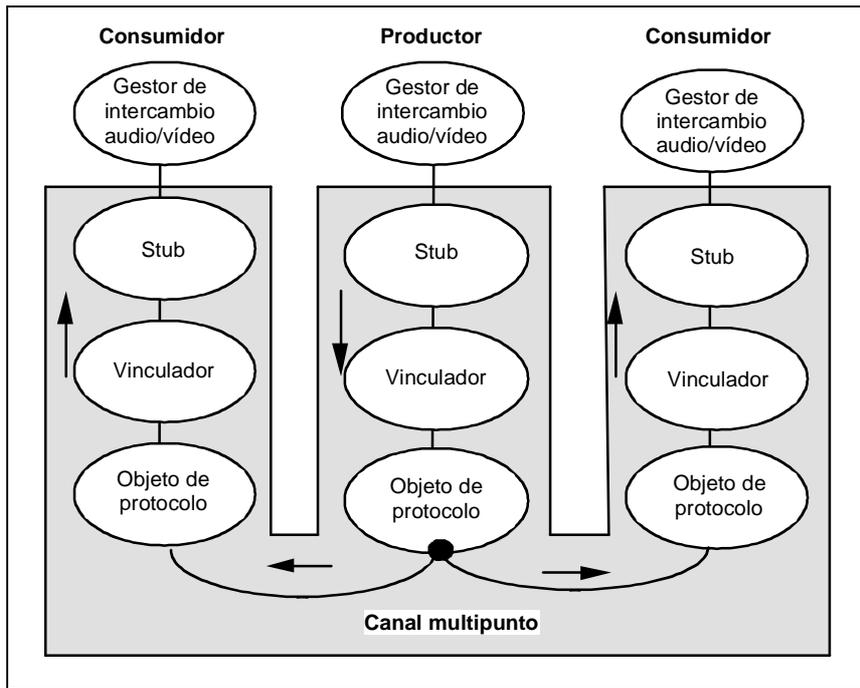


T0726430-96/d28

Figura 28 – Soporte de ingeniería

Los objetos de vinculación verifican la compatibilidad de las interfaces que han de enlazarse y mantienen la integridad de la vinculación entre los gestores del intercambio de audio/vídeo.

Los objetos de protocolo interactúan unos con otros para transportar la información necesaria para soportar interacciones transparentes a la distribución entre los gestores del intercambio de audio/vídeo. Los objetos de protocolo se eligen de modo que respeten las constricciones de calidad de servicio (por ejemplo, sobre tiempo real, seguridad).



T0726440-96/d29

Figura 29 – Canal multipunto para el tren 2

### 12.1.6 Especificación de tecnología

La especificación de tecnología de MMCS especifica la implementación del sistema en términos de los componentes de soporte físico y de soporte lógico. Los puntos de referencia identificados en las otras especificaciones de punto de vista se definen como puntos de conformidad en los cuales puede verificarse el comportamiento del MMCS.

La especificación de tecnología expresa también requisitos que deben cumplir las estaciones de trabajo para la representación de vídeo, audio y texto. Se requieren redes de alta velocidad y área ancha para satisfacer las necesidades de comunicación proporcionando una anchura de banda suficiente para transmitir vídeo/audio y texto a los participantes situados en lugares diferentes.

### 12.2 Vinculación de trenes audio/vídeo de múltiples partes

Este ejemplo se refiere a la vinculación de trenes multipartitos utilizada en el sistema presentado en el ejemplo precedente.

En este ejemplo se ha seguido el método de esbozar el dominio del problema en términos computacionales y proporcionar entonces las cinco especificaciones correspondientes de punto de vista.

Primeramente se definen algunos conceptos y reglas adicionales aplicables a este problema. En la especificación de empresa se introducen los roles de *interesados*<sup>1)</sup> (usuario, cliente, proveedor) para estructurar el dominio del problema con más detalle. Se utilizó OMT [Rumbaugh 91] e IDL como notaciones particulares para expresar las especificaciones de información y computacional, respectivamente.

Para la especificación de información se introducen varias relaciones entre clases del esquema invariante. Estas relaciones se derivan de un conjunto de relaciones básicas de OMT y usualmente están parametrizadas con texto para que su significado sea más preciso.

<sup>1)</sup> Interesado ("stakeholder") es un concepto de telecomunicaciones que indica una organización o persona que tiene un interés comercial o reglamentario en servicios de telecomunicaciones.

En la especificación computacional se introduce el controlador y despachador de audio/vídeo para manejar la vinculación de trenes de audio/vídeo de múltiples partes.

Para la especificación de ingeniería se introduce un canal de tren especializado para el transporte de flujos continuos. Se presenta una configuración de objetos de ingeniería para el soporte de la vinculación de trenes de audio/vídeo de múltiples partes, incluyendo objetos para controlar y coordinar múltiples canales de tren. Hubiera sido posible especificar el controlador y despachador de una manera distribuida, pero esta complejidad adicional no se introdujo en el ejemplo.

La implementación descrita en el punto de vista de la tecnología se utiliza para validar el proceso de modelado en los puntos de vista ODP. Es probable que, en un entorno operativo, se utilice un soporte físico/lógico más específico, por ejemplo para satisfacer los estrictos requisitos de calidad de funcionamiento que se imponen en las aplicaciones multimediales.

### 12.2.1 Descripción general

El intercambio de medios continuos en aplicaciones multimediales distribuidas es complejo. Por ejemplo, en una aplicación de conferencias multimediales en tiempo real, los participantes están geográficamente separados y se comunican intercambiando informaciones de vídeo y audio en tiempo real. El intercambio audiovisual debe ser lo más natural y flexible posible. Esto entraña la necesidad de tener en cuenta exigencias como la de la sincronización de los labios y la sincronización de la visualización en múltiples estaciones de trabajo, al especificar la aplicación de conferencias multimediales.

Para cumplir estos requisitos, la aplicación de conferencias multimediales impone requisitos estrictos de la calidad de funcionamiento y sincronización de la red para el intercambio de flujos de audio y vídeo. Además, el número de flujos intercambiados y la calidad correspondiente pueden cambiar en el curso de la conferencia. Esto se debe al hecho de que la aplicación proporciona operaciones para incorporarse a la conferencia y separarse de ella, y para modificar la calidad de servicio de los flujos.

Para tratar esta compleja funcionalidad, el RM-ODP define la noción de objeto de vinculación en el lenguaje computacional. El RM-ODP proporciona el concepto computacional teórico sin los refinamientos específicos requeridos en una determinada área de problema. En este ejemplo, el objeto de vinculación se especifica en términos de cinco especificaciones de punto de vista ODP de un determinado objeto de vinculación, es decir, se han proporcionado objetos de vinculación de audio/vídeo de múltiples partes [Gay 95]. Este objeto gestiona las interfaces de tren que se utilizan para las interacciones de audio/vídeo de múltiples partes en tiempo real. Pueden realizarse también operaciones de control sobre el objeto de vinculación multipartita.

La figura 30 muestra la representación computacional del objeto de vinculación de audio/vídeo de múltiples partes y su entorno.

El rectángulo en el centro indica el objeto de vinculación de audio/vídeo de múltiples partes. Su entorno (zonas sombreadas) está constituido por las partes aplicación y sistema de soporte, y la infraestructura de red de soporte. Los símbolos  $\perp$  indican a través de los cuales los productores y consumidores de audio/vídeo intercambian audio y/o vídeo (1). El objeto de vinculación de audio/vídeo de múltiples partes gestiona las interacciones entre las interfaces de tren que comprende. Encapsula los mecanismos utilizados para esto y hace abstracción de los aspectos de distribución. El símbolo  $\perp$  en la parte superior del rectángulo indica la interfaz de control de tren del objeto de vinculación. A través de esta interfaz, el objeto de vinculación de audio/vídeo de múltiples partes proporciona operaciones 3 y 4 al entorno que controla su funcionamiento.

### 12.2.2 Especificación de empresa

La especificación de empresa proporciona una descripción de los requisitos y objetivos que el entorno impone al sistema que va a diseñarse. Justifica el diseño de un sistema. Los conceptos de empresa de objeto de empresa que desempeñan roles de *acciones de realización* (dícese también acciones performativas) se utilizan para describir la vinculación de audio/vídeo de múltiples partes.

Los roles que los interesados en telecomunicaciones pueden desempeñar con respecto a un servicio (es decir, usuario, cliente y proveedor) se utilizan en la especificación del objeto de vinculación. Para la aplicación y el sistema de soporte, esto tiene por consecuencia la identificación de objetos de productor/consumidor de audio/vídeo (usuarios) y objetos que gestionan a estos usuarios (clientes). Para la infraestructura de red de soporte se distingue un objeto de gestión (proveedor) que gestiona el objeto de vinculación.

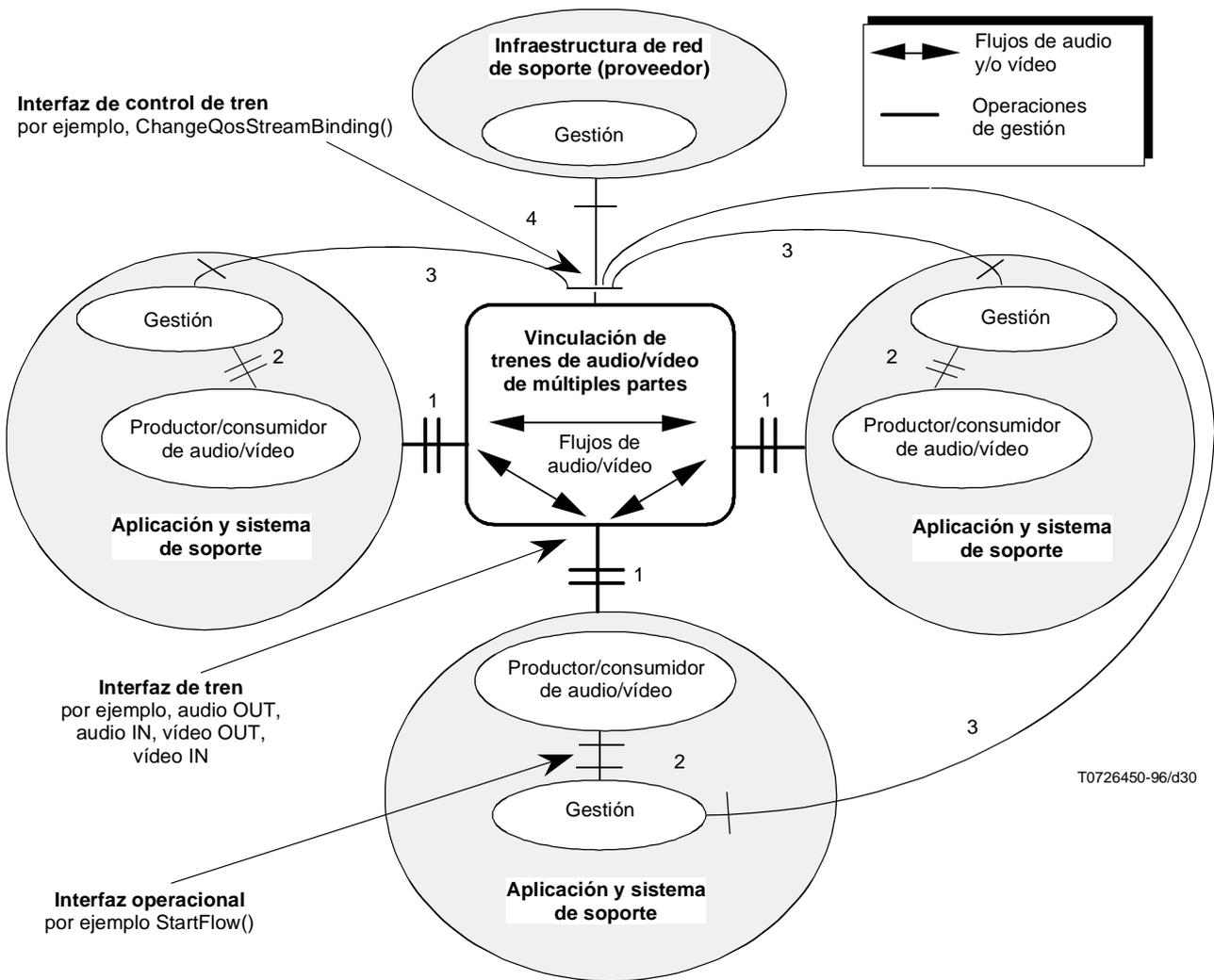


Figura 30 – Vinculación de trenes de audio/vídeo de múltiples partes

**12.2.2.1 Usuario: Productor/consumidor de flujos de audio/vídeo**

Un usuario es un objeto de empresa, en la aplicación y el sistema de soporte, que desempeña un rol cuyo objetivo es producir y/o consumir flujos a través de su interfaz de tren (véase la figura 30, 1). Son ejemplos de usuarios los micrófonos, altavoces, cámaras y pantallas. Un usuario indica el tipo de flujo que puede manejar y los formatos de codificación que requiere. Además, define los valores de los parámetros de calidad de servicio que requiere. Estos parámetros especifican la calidad de audio/vídeo en términos de televisión por radiodifusión, televisión de alta definición (HDTV), calidad audio de telefonía, alta fidelidad o calidad de disco compacto. Además de esto, se definen requisitos de calidad de servicio entre flujos, que prescriben, por ejemplo, que se requiere la sincronización de labios entre los flujos de audio y vídeo y que es deseable la entrega simultánea de audio/vídeo a múltiples usuarios.

**12.2.2.2 Cliente: Aplicación y sistema de soporte locales**

Un cliente es un objeto de empresa que desempeña un rol cuyo objetivo es gestionar y dar soporte al productor/consumidor de audio/vídeo. Tiene en cuenta las políticas de usuario que deberá soportar y gestionar. Los clientes pueden invocar acciones de realización (véase la figura 30, 2) para crear usuarios, suprimir usuarios, o ajustar políticas de usuarios existentes.

Un cliente ejecuta acciones de realización relativas a la configuración y a la gestión de los recursos. Por ejemplo, tiene que gestionar la prioridad de los flujos. En caso de congestión de la red o de problemas en recursos locales, el flujo de más baja prioridad se suprime o retrasa. El cliente realiza también (re)negociaciones de extremo a extremo con otros clientes para determinar valores aceptables, preferidos e inaceptables de los parámetros de calidad de servicio. Esto último se refleja en el 'contrato de vinculación' (véase la figura 31). Un contrato de vinculación describe el resultado convenido de la negociación entre el usuario, el cliente y el proveedor, que deberá seguirse durante la existencia del objeto de vinculación multipartita. El cliente puede ejecutar, sobre el objeto de vinculación, acciones de realización (véase la figura 30, 3) que se relacionan con la gestión de la vinculación de flujos de múltiples partes. Estas acciones se refieren por ejemplo al establecimiento, supresión y ajuste de una vinculación de tren de acuerdo con los requisitos de calidad de servicio de extremo a extremo (re)negociados. Las acciones de realización pueden también ser iniciadas por el objeto de vinculación para indicar, por ejemplo, que no puede mantener los valores negociados de calidad de servicio.

**12.2.2.3 Vinculación y proveedor**

La vinculación es un objeto de empresa que permite el intercambio de flujos de audio y vídeo entre usuarios geográficamente separados. El proveedor es un objeto de empresa que ofrece una infraestructura de conexiones que es utilizada por la vinculación y que gestiona la vinculación de acuerdo con el estado de la infraestructura.

El proveedor es responsable de la topología del flujo, aspectos de facturación, aspectos de seguridad, gestión de averías y calidad de servicio proporcionada por la red y recursos subyacentes. Incumbe al proveedor la gestión de la red, de modo que el contrato de vinculación quede garantizado. Por ejemplo, selecciona un encaminamiento apropiado para el canal de audio/vídeo y reserva recursos en cada nodo en esa ruta. Si el proveedor deja de estar en condiciones de garantizar el contrato de vinculación, ejecuta acciones (véase la figura 30, 4) que afectan a la vinculación con respecto a la (re)negociación del contrato de vinculación.

**12.2.3 Especificación de información**

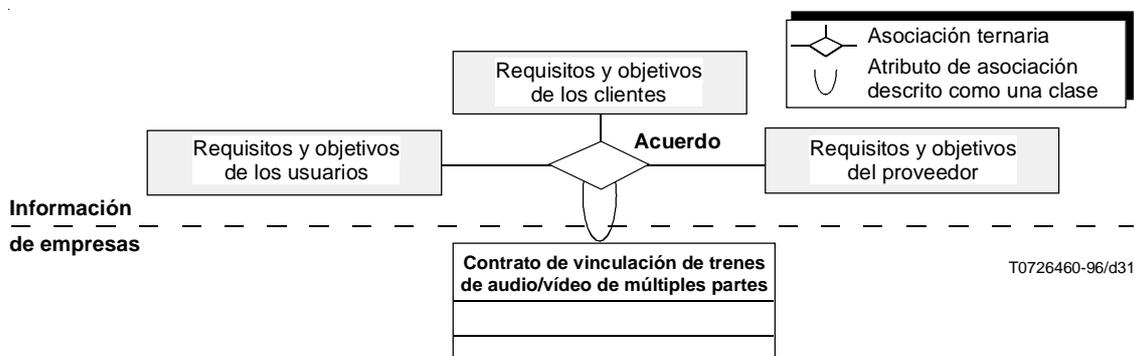
La especificación de información de la vinculación de tren de audio/vídeo de múltiples partes describe la información pertinente a los interesados en la vinculación. Tiene en cuenta los requisitos y objetivos indicados en la especificación de empresa.

**12.2.3.1 De especificación de empresa a especificación de información**

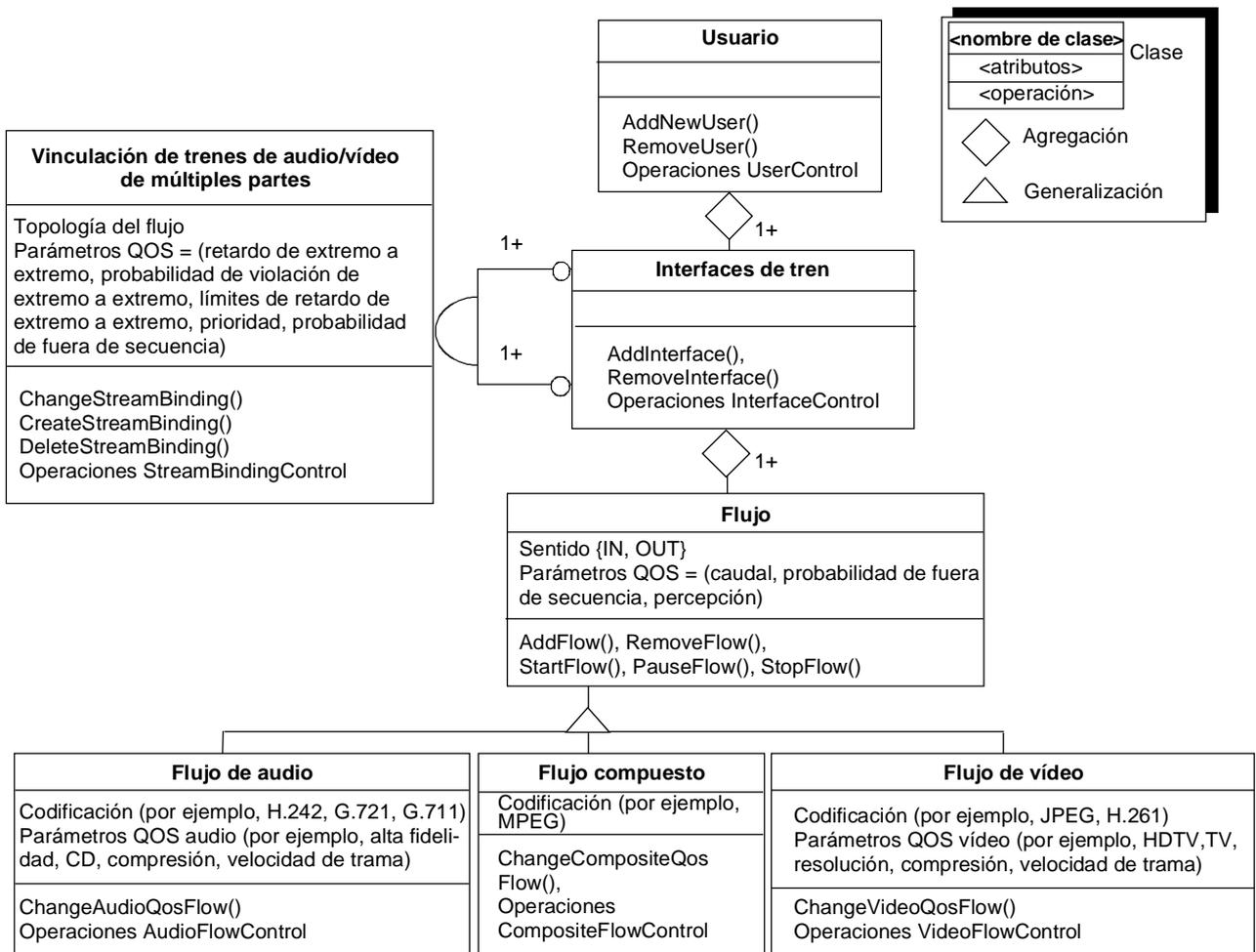
La información relevante para los usuarios, clientes y proveedor se especifica en el contrato de vinculación. El contrato es el resultado de un acuerdo entre usuarios, clientes y proveedor. Satisface sus requisitos y objetivos consignados en la especificación de empresa. La figura 31 muestra la relación entre la especificación de empresa y la especificación de información según la técnica OMT.

**12.2.3.2 Esquema invariante del contrato de vinculación**

El contrato de vinculación, como se muestra en la figura 31, se considera un atributo entre interesados y se describe como una clase única. Sin embargo, en un nivel más detallado, la especificación de información es más compleja. La figura 32 muestra el esquema invariante del contrato de vinculación de trenes de audio/vídeo de múltiples partes.



**Figura 31 – Contrato de vinculación entre usuarios, clientes y proveedor**



T0726470-96/d32

**Figura 32 – Esquema invariante de contrato de vinculación**

La estructura común a todos los contratos entre clientes, usuarios y proveedor se especifica como un esquema invariante. En el caso de un contrato de vinculación, contiene información sobre los usuarios en la vinculación (objetos de información de usuario), las interfaces de tren que intervienen en la vinculación (objetos de información de interfaz de tren), y las operaciones que los interesados pueden invocar. La información en la vinculación se modela por medio del objeto de información de vinculación de trenes de audio/vídeo de múltiples partes. El esquema invariante para este objeto de información especifica todas las operaciones que los interesados pueden invocar.

Un usuario puede tener una o más interfaces de tren, lo que implica que cada objeto de información de usuario consta de uno o más objetos de información de interfaz de tren. Una interfaz de tren comprende uno o más flujos, como resultado de lo cual un objeto de información de interfaz de tren consta de uno o más objetos de información de flujo (de audio, vídeo o compuesto). Un objeto de información de flujo consiste en atributos que indican, entre otras cosas, el sentido de los flujos y los parámetros de calidad de servicio. Los parámetros de calidad de servicio definidos en la especificación de información serán manipulados en la especificación computacional.

La información relativa a la vinculación es captada por el objeto de información de vinculación de trenes de audio/vídeo de múltiples partes. Este objeto de información relaciona dos o más objetos de información de interfaz de tren. Contiene información sobre la topología del flujo y describe la calidad de servicio que debe mantenerse cuando se intercambian flujos de audio/vídeo entre las interfaces.

### 12.2.3.3 Esquema estático del contrato de vinculación

Un contrato de vinculación específico que existe entre interesados en cualquier momento se especifica como un esquema estático. Un contrato de vinculación debe satisfacer su esquema invariante y consta de objetos de información de usuario, objetos de información de interfaz de tren, objetos de información de flujo, y objetos de información de vinculación de trenes de audio/vídeo de múltiples partes. En el contrato puede haber otros objetos de información, siempre que no estén en contradicción con el esquema invariante.

### 12.2.3.4 Esquema dinámico del contrato de vinculación

Un esquema dinámico para un contrato de vinculación tiene dos aspectos: en primer lugar, el efecto que la invocación de operaciones produce sobre el contrato y, en segundo lugar, las condiciones en que los interesados pueden invocar estas operaciones. A continuación se examinan ambos aspectos.

El efecto de las operaciones sobre la información del contrato de vinculación depende mucho de las opciones de implementación elegidas. En general, los efectos pueden clasificarse en tres categorías: efecto de notificación, efecto de negociación y efecto nulo.

Una operación que tiene un efecto de notificación es aquella en que un interesado informa el contrato de vinculación de un objeto computacional de nueva creación. El efecto del contrato es la adición de un nuevo objeto de información. Por ejemplo, la operación AddNewUser (añadir nuevo usuario) produce un nuevo objeto de información de usuario, uno o más objetos de información de interfaz de tren, y uno o más objetos de información de flujo.

Una operación que tiene un efecto de negociación es aquella en que un interesado negocia con el objeto de vinculación, y con otros interesados, sobre un cambio en el contrato de vinculación. Si la negociación es exitosa, se hacen cambios en el contrato de vinculación. Por ejemplo, una operación RemoveUser (suprimir usuario) exitosa tiene por consecuencia la supresión de un objeto de información de usuario y de los correspondientes objetos de información de usuario e información de interfaz de tren. Además, se modificará o suprimirá el objeto de información de vinculación que relaciona las interfaces del usuario suprimido con las interfaces de otros usuarios.

Una operación que produce un efecto nulo es aquella que no afecta a los objetos de información en el contrato de vinculación. Estas operaciones son principalmente operaciones de control. Por ejemplo, la operación PauseFlow (detener temporalmente el flujo) tendrá por efecto una detención temporal del flujo. Esto no influirá en la información del contrato de vinculación.

El esquema dinámico describe también condiciones para invocar operaciones sobre el contrato de vinculación. Los esquemas invariantes y estáticos no imponen un orden específico de las operaciones que los interesados puedan invocar. No obstante, para obtener un contrato de vinculación significativo es necesario definir condiciones con respecto a la invocación de operaciones. A continuación se indican algunos ejemplos:

- Un cliente sólo puede invocar una operación AddNewUser (añadir nuevo usuario) en una vinculación de tren existente.
- Los clientes sólo pueden invocar las operaciones ChangeAudioQoSFlow (cambiar flujo de QOS audio) y RemoveFlow (suprimir flujo) en flujos existentes.
- Los clientes y proveedores solo pueden invocar una operación ChangeStreamBinding (cambiar vinculación de tren) o DeleteStreamBinding (suprimir vinculación de tren) en una vinculación existente.

## 12.2.4 Especificación computacional

En esta subcláusula se especifica la firma de la interfaz de control de vinculación de tren. Para tener una descripción completa de la interfaz se debe describir también los contratos de comportamiento y entorno.

### 12.2.4.1 Correspondencia de objeto de información a objeto computacional

La correspondencia entre objetos de información y objetos computacionales no es necesariamente una correspondencia de uno a uno. Las correspondencias entre la especificación de información y la especificación computacional tienen que especificarse en cada caso, de modo que pueda determinarse la consistencia entre las especificaciones. La agrupación de clases en objetos es una decisión que toma el diseñador del servicio. En esta etapa no es necesario tener en cuenta los aspectos de distribución.

La figura 33 muestra la correspondencia de varias clases de información con interfaces computacionales. Para la interfaz de control (véase la figura 33, 3 y 4) de la vinculación de trenes de audio/vídeo de múltiples partes se tienen en cuenta las operaciones de la clase vinculación de trenes de audio/vídeo de múltiples partes. La interfaz de operación del productor/consumidor de audio/vídeo 2, se reflejará en las operaciones definidas en la clase flujo. La interfaz de tren, 1, tiene las características de los atributos de la subclase flujo. Los atributos especificados en la especificación de información serán representados por parámetros en operaciones computacionales. Las denominaciones de las operaciones y los parámetros en ambos puntos de vista son independientes.

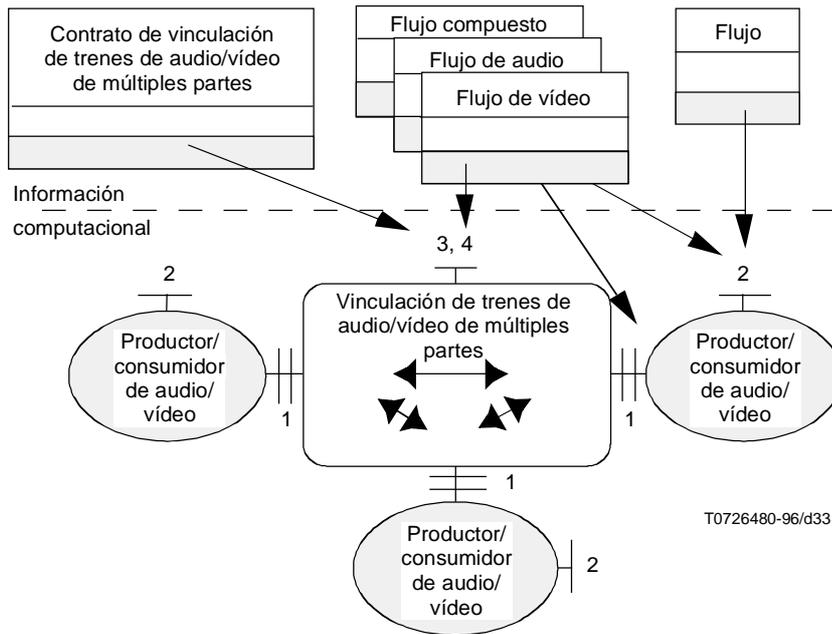


Figura 33 – Correspondencia de objeto de información a objeto computacional

#### 12.2.4.2 Especificación en IDL

NOTA – Lo siguiente está estrechamente relacionado con OMG-IDL [OMG IDL].

El RM-ODP describe un modelo computacional que puede utilizarse para aplicaciones distribuidas, pero no proporciona un lenguaje de especificación específico para objetos e interfaces computacionales. Por eso se utiliza aquí un lenguaje de especificación adicional, IDL, para derivar una especificación computacional de la interfaz de control de vinculación de tren (véase el cuadro 1). El lenguaje IDL proporciona los medios para formular especificaciones computacionales orientadas hacia las telecomunicaciones y multimedia. Las especificaciones IDL derivadas se basan en la especificación de información.

Cuadro 1 – Especificación IDL de la interfaz de control de vinculación de tren

```

interface template StreamBindingControlInterface; /* (3), (4) operation interface type*/

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);

/* Son posibles operaciones de control de vinculación de tren adicionales, determinadas por las partes, aplicación, sistema
y gestión de red. */

/* behaviour (comportamiento)' Un ejemplar (instancia) de esta plantilla proporciona otros objetos computacionales para
ejecutar acciones de control sobre el objeto de vinculación de múltiples partes.' */
    
```

### 12.2.4.3 Opción computacional para la configuración de intercambio de audio/vídeo de múltiples partes

La vinculación de trenes de audio/vídeo de múltiples partes puede refinarse para conseguir una configuración de ingeniería. Son posibles varias soluciones, pero la que se haya elegido se utiliza en muchos sistemas multimedia distribuidos. Varias implementaciones de sistemas multipartitos distribuidos tienen un componente funcional, llamado el controlador y despachador de audio/vídeo, que gestiona los flujos de audio/vídeo. Recibe todos los flujos de audio/vídeo de los productores y (tras una posible manipulación) los refleja a todos los consumidores. Este método se ha adaptado para utilizarlo en la vinculación de trenes de audio/vídeo de múltiples partes, como se muestra en la figura 34.

El controlador y despachador de audio/vídeo se encarga de redireccionar las operaciones de control de trenes 3 y 4 a cada vinculación de subtren, 5. Se encarga también del establecimiento, control y liberación de las vinculaciones de audio/vídeo entre los productores y los consumidores. Negocia los requisitos y objetivos identificados en la especificación de empresa (por ejemplo, algoritmos de codificación y compresión, velocidad de trama). Una vinculación de trenes de audio/vídeo enlaza cada uno de los objetos de productor/consumidor de audio/vídeo con el controlador y despachador de audio/vídeo.

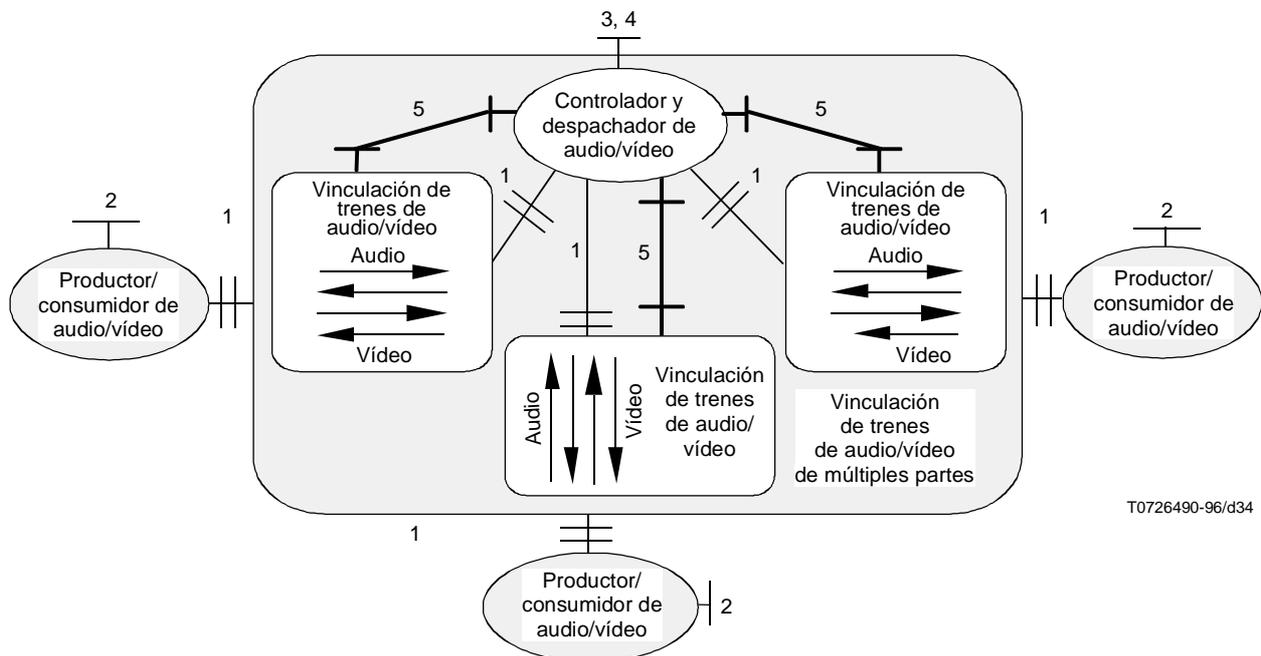


Figura 34 – Objetos que intervienen en un intercambio de audio/vídeo de múltiples partes

### 12.2.5 Especificación de ingeniería

Esta subcláusula proporciona una especificación de ingeniería del objeto de vinculación de trenes de audio/vídeo de múltiples partes definido en la especificación computacional.

#### 12.2.5.1 De especificación computacional a especificación de ingeniería

Para ejecutar una especificación computacional hay que hacerla corresponder con una especificación de ingeniería. Esta especificación de ingeniería mantiene el comportamiento descrito en la especificación computacional.

La transformación de la especificación computacional a la de ingeniería, de una interfaz de tren (véase la figura 35, 1) conduce a la creación de un canal de tren de productor/consumidor especializado para flujos continuos. Los parámetros de calidad de servicio asociados con el flujo de audio/vídeo definido en la especificación computacional influyen en la elección de los componentes del canal de tren.

Una interfaz de operación (véase la figura 35, 2) se refleja en la especificación de ingeniería como una configuración de canal cliente-servidor, definida en el RM-ODP. Los contratos de entorno específicos para las interfaces se tienen en cuenta cuando se establece un canal entre los objetos computacionales considerados.

La *interfaz de control de vinculación de tren* computacional (véase la figura 35, 3, 4 y 5) está situada en diferentes nodos en la representación de ingeniería. La comunicación de estos nodos (no se representa en la figura 35) se efectúa a través de canales de control que tienen la misma estructura que los canales de operación. Además, se puede crear objetos de soporte para (por ejemplo, un objeto de sincronización) para gestionar y controlar un conjunto de canales de tren interrelacionados.

El objeto de *productor de audio/vídeo*, el objeto de *consumidor de audio/vídeo* y el objeto de *controlador y despachador de audio/vídeo* se transforman en objetos de ingeniería básicos. Si estos objetos están distribuidos en diferentes nodos, hay que proceder a una ulterior descomposición y hay que crear varios objetos de ingeniería y partes canal. La figura 35 muestra la correspondencia de dos productores y consumidores de audio/vídeo computacionales en una especificación de ingeniería.

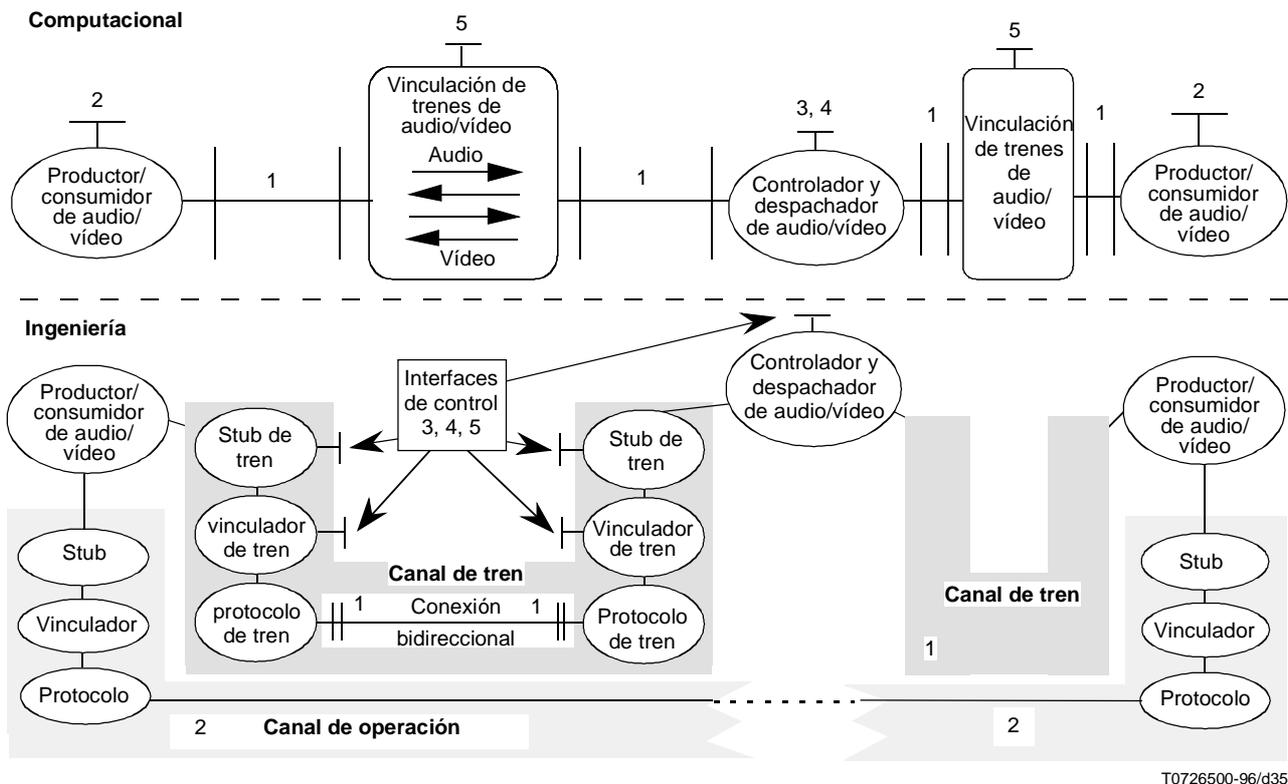


Figura 35 – Correspondencia de especificación computacional a especificación de ingeniería

### 12.2.5.2 Canal de tren de ingeniería

El concepto de canal ODP proporciona los mecanismos de ingeniería para asegurar interacciones transparentes de distribución entre objetos de ingeniería básicos. El canal consta de tres objetos de ingeniería, a saber: los objetos de protocolo, vinculador y stub. Dos tipos diferentes de información se transportan a través de los canales. En primer lugar se transportan las operaciones de control que permiten, por ejemplo, la negociación de la calidad de servicio. Estas operaciones necesitan poca anchura de banda, pero exigen una gran fiabilidad. En segundo lugar se transportan las interacciones en tiempo real, como los intercambios de voz y vídeo, que necesitan una gran anchura de banda, pero su fiabilidad puede ser menor. En consecuencia, los canales se dividen en un canal de operación y un canal de tren, cada uno de los cuales tiene sus propias características. En esta subcláusula se presentan primeramente los conceptos de los objetos de stub, vinculador y protocolo, y seguidamente se describe la manera de establecer un determinado canal de tren.

El **objeto de stub** proporciona funciones de adaptación para soportar la interacción entre interfaces de objeto de ingeniería básico en diferentes nodos. Para las invocaciones de operaciones, el objeto de stub proporciona la activación/desactivación de parámetros de operación para permitir operaciones transparentes de acceso. Los trenes requieren una funcionalidad diferente del objeto de stub debido a la diferente naturaleza de las informaciones que se intercambian. Este objeto deberá proporcionar el mecanismo para codificar y decodificar información de audio/vídeo. Además, los datos disponibles para el productor o consumidor de audio/vídeo deben notificarse y los objetos de stub de tren proporcionan operaciones de control a recursos locales (por ejemplo, aumentar el tamaño de una memoria tampón) y la notificación de eventos relativos al tren (por ejemplo, notificación de una modificación de la calidad de servicio, ausencia de espacio disponible para memoria tampón, salida de un dato, etc.). Un objeto de stub tiene una interfaz de presentación para uso por el objeto que está vinculado al canal, y una interfaz de control por ejemplo para gestión de la calidad de servicio.

Los **vinculadores** interactúan unos con otros para mantener la integridad de la vinculación. Se mantiene información sobre el canal. Los vinculadores son también responsables de la validación de la referencia de interfaz y de la interacción con el objeto de reubicador para recuperar información sobre la ubicación de la interfaz tras un error de vinculación. En el caso de trenes, se mantiene información con respecto a la calidad de servicio requerida. Un vinculador tiene una interfaz de control que permite efectuar cambios en la configuración del canal y la destrucción de la totalidad o de una parte del canal.

El **objeto de protocolo** asegura que los objetos computacionales puedan interactuar a distancia con cada uno de los demás. Se necesitan objetos de protocolo si los objetos computacionales que deberán vincularse están situados en diferentes nodos. En general, se utiliza el mecanismo de llamada de procedimiento distante (RPC) cuando se envía una operación en forma de mensaje a un objeto de protocolo distante capaz de recibirlo. El objeto a que está destinada la llamada ejecuta el procedimiento y devuelve un mensaje de respuesta. Para la vinculación de trenes de audio/vídeo de múltiples partes, el tipo RPC de objetos de protocolo es adecuado para los objetos computacionales que hacen invocaciones de operaciones, por ejemplo para el canal de control/gestión (véase la figura 35, 3, 4 y 5). Sin embargo, para el intercambio de flujos continuos es necesario otro protocolo especializado (denominado protocolo de tren) que no tiene el mecanismo RPC. La RPC requiere que cada elemento de los datos almacenados en una memoria tampón que vaya a transferirse se asocie a una acción distinta, es decir, no existe una relación entre las llamadas de procedimiento distante ya efectuadas y las futuras. Los flujos continuos requieren relaciones entre las llamadas y se aplica un protocolo de tren que crea un canal virtual entre dos objetos de protocolo y lo mantiene durante el intercambio de flujos de audio/vídeo. En este caso se puede definir específicamente relaciones entre datos de audio/vídeo. Los objetos de protocolo pueden interactuar con objetos fuera de un canal (por ejemplo con un comerciante para obtener la información que necesitan).

### 12.2.6 Especificación de tecnología

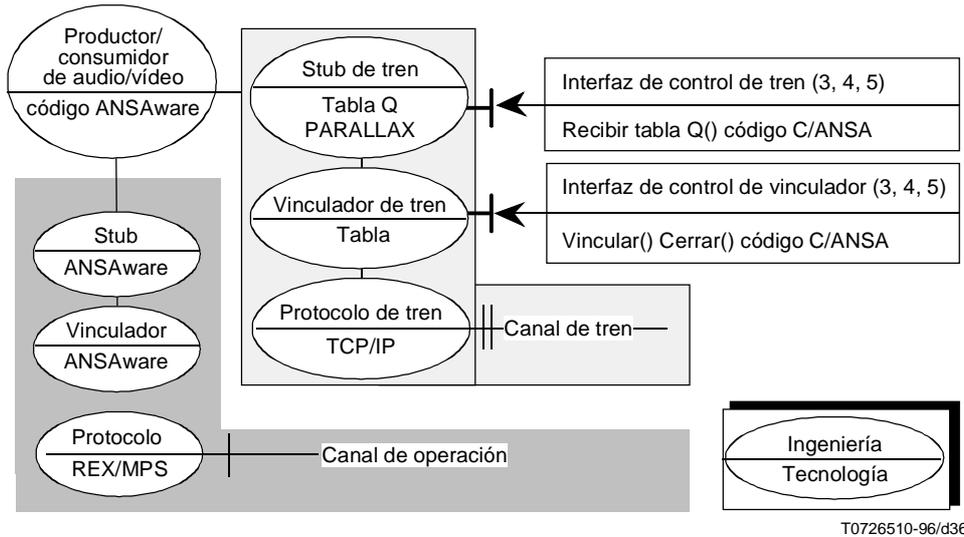
En esta subcláusula se describe, mediante la utilización de ANSAware, una implementación inicial de una vinculación de tren que se basa en una arquitectura similar a la del ODP. Se utiliza ANSAware como una plataforma distribuida que tiene la ventaja de ofrecer varios mecanismos de ingeniería realizados (por ejemplo, ANSAware soporta interfaces de operación). La especificación de tecnología presentada corresponde a una implementación de un servicio videofónico multipartito que permite a los usuarios de extremo intercambiar información audiovisual a través de su computador de mesa. Se puede incorporar/separar dinámicamente usuarios a/de una sesión en curso.

La especificación de tecnología está determinada por las otras especificaciones de punto de vista, pero la implementación se basa principalmente en la especificación de ingeniería. La especificación de tecnología consiste en una descripción del soporte físico y del soporte lógico que pueden implementar la especificación de ingeniería teniendo en cuenta requisitos de empresa adicionales (por ejemplo, qué soporte físico y qué soporte lógico están disponibles).

La figura 36 muestra una correspondencia entre objetos de ingeniería y soluciones tecnológicas. Se concentra en la implementación del canal de tren y del canal de operación, y muestra cómo los objetos de ingeniería se realizan en componentes de soporte físico y de soporte lógico.

## 12.3 Ejemplo de gestión – Objeto métrico

L'ISO/CEI 10164-11 (Metric Objects and Attributes) (Objetos y atributos métricos) especifica un objeto métrico de monitor de la media, que explora el mismo valor de atributo a partir de un objeto observado identificado, a intervalos de tiempo regulares [periodo de granularidad (GP, *granularity period*)], actualiza una estimación del promedio móvil del valor de atributo observado, y aplica un mecanismo de umbral basado en el valor promedio de la media, que provoca la emisión de una notificación cuando se excede el umbral.



**Figura 36 – Correspondencia de especificación de ingeniería a especificación de tecnología**

### 12.3.1 Especificación de empresa

#### Comunidades:

Comunidad de operación métrica – comprende los siguientes roles:

- Rol de explorador – Rol de objeto de cliente que se encarga de iniciar exploraciones de un objeto observado asociado que tiene el rol observado.
- Rol observado – Rol de objeto de servidor que corresponde a una exploración de un valor de atributo observado.

Comunidad de control métrico – comprende los siguientes roles:

- Rol de controlador métrico – Rol de objeto de cliente para gestión objeto, que se encarga de iniciar operaciones de control métrico para uno o más objetos métricos.
- Rol de control métrico – Rol de objeto de servidor para objeto métrico que responde a operaciones de control para cambiar el estado del objeto métrico correspondiente a los parámetros de algoritmo asociados con un comportamiento métrico.

Comunidad de notificación métrica – comprende los siguientes roles:

- Rol de notificador métrico – Rol de objeto de cliente de objeto métrico que emite notificaciones relativas al algoritmo métrico que opera sobre valores de atributos observados explorados.
- Rol de distribución de notificaciones – Rol de objeto de servidor que recibe notificaciones emitidas para distribuirlas subsiguientemente.

#### Políticas:

Política de explorador – el objeto explorador debe iniciar exploraciones a intervalos regulares, establecidos por el atributo de control del periodo de granularidad.

Política de observación – el objeto observado debe responder con el valor observado dentro de un periodo de granularidad que corresponde al momento en que se recibe la petición de exploración. Si no puede responder, se señala un error y se invalida la exploración para esa observación.

Política de control métrico – el objeto métrico debe reflejar los cambios de los parámetros de algoritmo efectuados a través del rol de control métrico en el comportamiento de procesamiento de su algoritmo métrico de promedio móvil. Hay que controlar los siguientes parámetros de algoritmo: periodo de granularidad, periodo de tiempo móvil, puntero de la relación de instancia del objeto observado, identificador del atributo observado, umbral de disparador de notificación, umbral de rearme.

Política de notificación métrica – el objeto métrico debe emitir una notificación cuando el promedio móvil del valor observado rebasa el umbral de disparador de notificación. Un mecanismo de histéresis proporciona un umbral de rearme, que, al ser rebasado en sentido opuesto, rearma el umbral superior para notificaciones subsiguientes.

### 12.3.2 Especificación de información

Los objetos de información de más interés son el objeto métrico y el objeto observado.

`meanMonitorMetricObject InfoDescription {`

**Atributos:**

**Id de atributo observado** – id utilizado para obtener el valor de atributo observado.

**Periodo de granularidad** – tiempo que transcurre entre exploraciones sucesivas del valor de atributo observado del objeto observado conexo.

**Periodo de tiempo móvil** – longitud efectiva del tiempo durante el cual se calcula el promedio móvil del valor de atributo observado.

**Calibre derivado** – el valor actual del promedio móvil obtenido por la exploración del objeto métrico y el comportamiento de actualización del algoritmo.

**Umbral de disparador de notificación** – valor que ha de compararse con el calibre derivado al final de cada actualización del algoritmo, emitiéndose una notificación cuando el disparador está armado y el valor del calibre derivado es mayor que este umbral. Una vez emitida la notificación se desarma el disparador, el cual quedará desarmado hasta que sea ulteriormente rearmado mediante un mecanismo de histéresis.

**Umbral de rearme** – valor que ha de compararse con el calibre derivado al final de cada actualización del algoritmo, rearmándose el disparador de notificación se rearma cuando el valor del calibre derivado es menor que este umbral.

**Estados:**

**Estado operacional** – refleja si el objeto métrico está o no funcionando.

**Estado de la exploración** – los valores son:

**espera** (del comienzo del siguiente periodo de granularidad)

**exploración** (espera del resultado de la exploración del objeto observado)

**Estado de arme/desarme del umbral** – los valores son armado, desarmado.

**Invariantes:**

El periodo de tiempo móvil debe ser mayor que el periodo de granularidad.

El valor del umbral de rearme debe ser menor que el valor umbral del disparador de notificación.

**Transiciones de estado:**

Cuando el estado operacional está habilitado pueden producirse las siguientes transiciones de estado de la información:

**espera -> exploración** – se produce cuando el objeto métrico alcanza el siguiente periodo de granularidad.

**exploración -> espera** – se produce:

a) cuando se ha retornado el valor de atributo observado y el algoritmo ha actualizado el valor derivado y ha aplicado el procesamiento de umbral; o

b) cuando se invalida la exploración.

**armado -> desarmado** – se produce cuando el procesamiento de umbral detecta un rebasamiento del umbral del disparador de notificación (lo que provoca la emisión de una notificación).

**desarmado -> armado** – se produce cuando el procesamiento de umbral detecta un rebasamiento del umbral de rearme.

**Relaciones:**

**Instancia de objeto observado – un puntero al objeto observado cuyo identificador de atributo observado se explora para cada periodo de granularidad.**

}

El objeto de información "objeto observado" es cualquier objeto que tenga un atributo identificable cuyo valor se pueda explorar. El único estado de interés para el objeto observado es el valor actual de sus atributos observados.

**observedObject InfoDescription {**

**Atributos:**

**atributo observado – cualquier atributo identificado que tenga un tipo de valor entero o real.**

}

El siguiente objeto de información modela el contenido de un objeto de registro de alarma de servicio asociado con un evento de rebasamiento, en sentido positivo, del umbral de disparador de notificación.

**qualityOfServiceAlarmRecord InfoDescription {**

**Atributos:**

**identificador de instancia de objeto métrico**

**identificador de instancia de objeto observado**

**identificador de atributo observado**

**notifyTime (notificar tiempo)**

**derivedGaugeValue (valor de calibre derivado)**

}

### 12.3.3 Especificación computacional

La distribución de un objeto métrico en un sistema diferente del objeto observado está fuera del ámbito de la norma sobre la función de gestión de sistemas de objetos métricos.

El objeto computacional y las plantillas de interfaz siguientes demuestran la descomposición de objetos computacionales para permitir una exploración distribuida de un objeto observado, por un objeto métrico.

**meanMonitorMetric ComputationalObject {**

**Interfaces de servidor:**

**meanMonitorControl\_Server**

**Interfaces de cliente:**

**scanObservedObjectValue\_Client**

**qualityOfServiceAlarm\_Client**

**BEHAVIOUR – véase ISO/CEI 10164-11**

}

**Plantillas de interfaz:**

**meanMonitorControl Interface {**

**OPERACIONES (atributos, todos tienen métodos de obtención y sustitución);**

**instancia de objeto observado**

**identificador de atributo observado**

**periodo de granularidad**

**periodo de tiempo móvil**

calibre derivado

umbral de disparador de notificación

umbral de rearme

estado operacional

Contrato de entorno:

Las operaciones de control tienen que influir en el comportamiento parametrizado del algoritmo métrico para las exploraciones que se originan después de establecidos los atributos.

}

scanObservedObjectValue Interface {

**OPERACIONES:**

el cliente invoca una operación de obtención sobre la instancia de objeto observado para obtener el valor actual del atributo observado.

Contrato de entorno:

Si como resultado de la operación de exploración no se ha devuelto el valor del objeto observado antes del comienzo del periodo de granularidad siguiente, la exploración se invalida y no podrá utilizarse para el proceso de actualización del algoritmo métrico.

}

qualityOfServiceAlarm Interface {

**OPERACIONES:**

"ISO/CEI 10164-4":qualityOfServiceAlarmNotification

Contrato de entorno:

Los valores de la notificación de alarma de calidad de servicio se establecen con los valores indicados en la especificación de objeto de información para el registro de alarma.

}

## 12.4 Ejemplo de base de datos

Este ejemplo se basa en un sistema de tramitación de pedidos (muy simplificado), que soporta las actividades mercantiles de una organización vendedora (o de una parte de una organización más compleja) encargada de suministrar ciertos productos a sus clientes.

La organización vendedora tienen varios almacenes que se utilizan para depositar los productos. Estos productos se venden a clientes, que son otras organizaciones que han sido aceptadas como asociados comerciales dignos de crédito. Una venta se desarrolla de la manera siguiente: un cliente hace un pedido de una determinada cantidad de uno o más productos; los productos solicitados se suministran al cliente de un almacén; el costo de los productos se factura al cliente; por último, se registra la recepción del pago efectuado por el cliente.

### 12.4.1 Especificación de empresa

Este sistema puede identificar los siguientes objetos de empresa:

- organización vendedora: una comunidad facultada para comerciar;
- enlace de cliente: una entidad, perteneciente a la organización vendedora a la que los clientes pueden hacer pedidos;
- gestión de almacén: una entidad perteneciente a la organización vendedora, encargada de la explotación de un almacén;
- cuentas: una entidad perteneciente a la organización vendedora, encargada de los asuntos financieros;
- tramitación de pedidos: una entidad perteneciente a la organización vendedora, encargada de la tramitación de los pedidos y de llevar los registros correspondientes;
- cliente: una organización que comercia con la organización vendedora (le hace pedidos).

En la figura 37 se presentan ejemplos de estos objetos de empresa.

**12.4.2 Especificación de información**

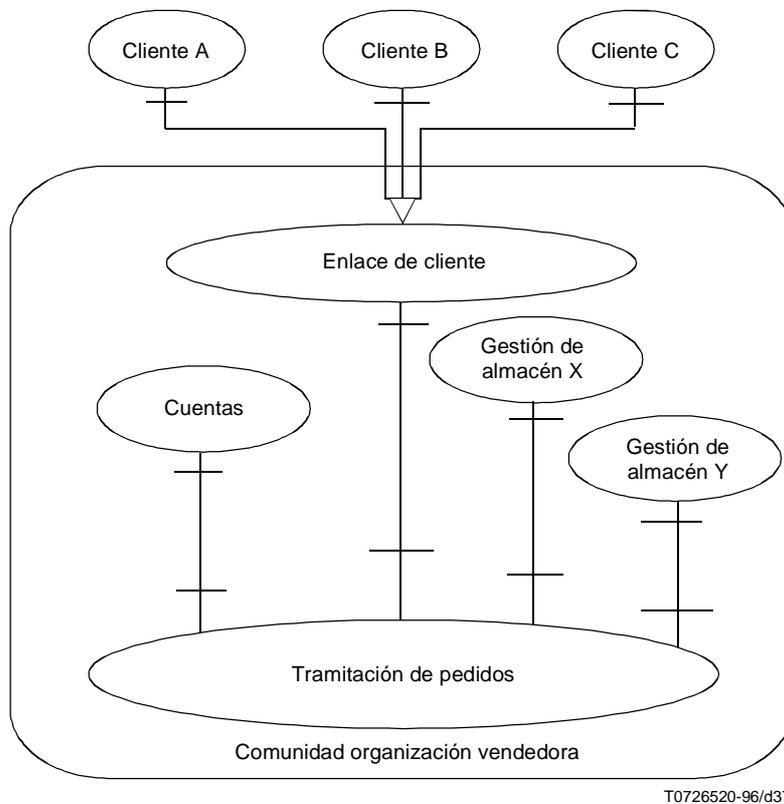
En la figura 38 se muestra un posible esquema invariante para este ejemplo, expresado mediante la notación OMT, con atributos y operaciones indicados a título de ejemplo. Representa el estado, comportamiento y relaciones entre los objetos de información relevantes para la organización vendedora. En particular, un objeto de información de cliente representa la información sobre el objeto de empresa de cliente que necesita la organización vendedora para sus actividades comerciales con las organizaciones clientes.

**12.4.3 Especificación computacional**

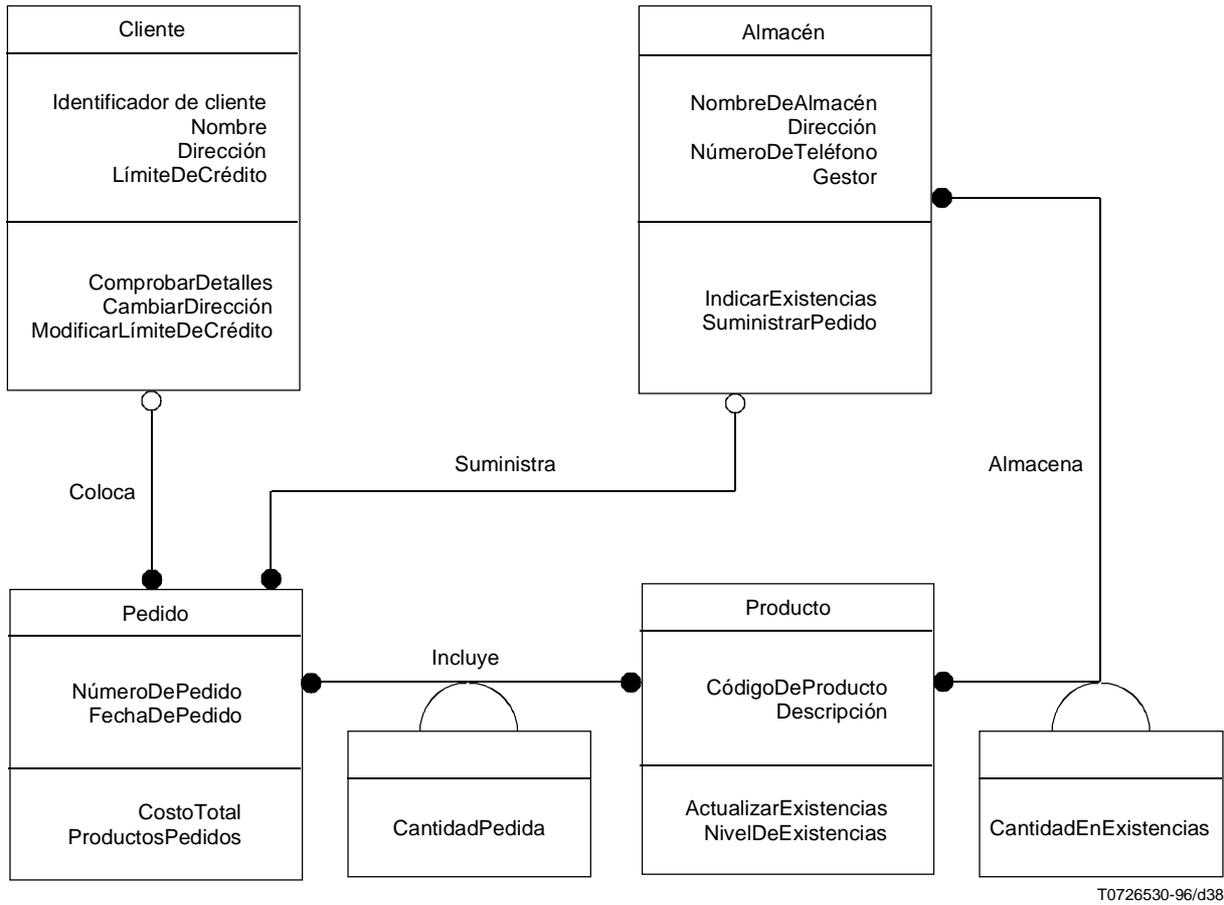
La especificación computacional para este ejemplo tiene dos características esenciales:

- hay muchos usuarios, cada uno de los cuales corresponde a uno de los objetos de empresa (enlace de cliente, gestión de almacén, cuentas), y cada uno de ellos solicita servicios que se relacionan con alguna parte del esquema de información;
- todas las apariciones del esquema de información deben tener una representación compartida y persistente.

Estos dos requisitos pueden satisfacerse mediante un sistema de base de datos; en la figura 39 se muestra un posible modelo computacional para este ejemplo.

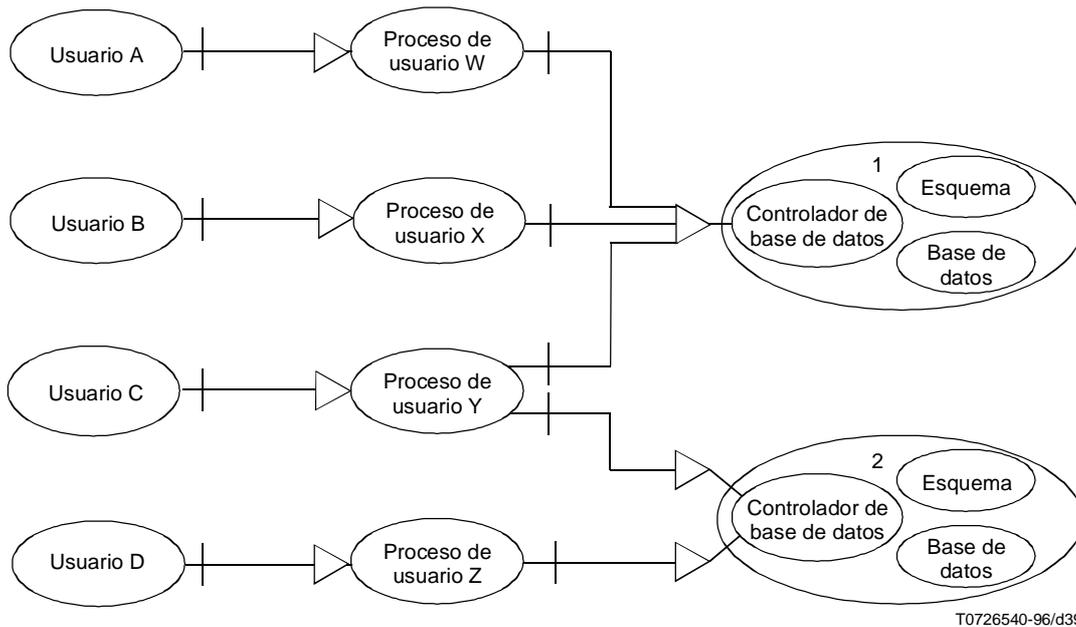


**Figura 37 – Sistema de tramitación de pedidos – Especificación de empresa**



T0726530-96/d38

Figura 38 – Sistema de tramitación de pedidos – Especificación de información



T0726540-96/d39

Figura 39 – Sistema de tramitación de pedidos – Especificación computacional

En este modelo computacional, cada uno de los objetos de usuario es de un tipo que corresponde o bien al de enlace de cliente, o al de gestión de almacén, o al de cuentas. Cada uno de los procesos de usuario proporciona servicios apropiados a su tipo de usuario cliente, lo que comprende algunos de los objetos de esquema de información, pero es improbable los comprenda todos. Todas las apariciones del esquema de información (pero no sus operaciones) están contenidas en los objetos de base de datos, y se puede aplicar políticas diferentes para su asignación entre las diferentes bases de datos, por ejemplo:

- atribución de objetos de acuerdo con el tipo, por ejemplo todos los objetos de cliente y de pedido en una base de datos y todos los objetos de almacén y de producto en la otra;
- atribución de objetos de acuerdo con alguna propiedad, por ejemplo, una base de datos mantiene instancias de todos los tipos de objeto para un almacén (digamos, un almacén de productos eléctricos) y otra base de datos que mantiene instancias de todos los tipos para otro almacén (digamos, un almacén de materiales de construcción);
- estas dos políticas pueden combinarse con la replicación de algunos objetos para responder a las diversas clases de exigencias de la tramitación (una ulterior descomposición es también posible).

## **Anexo A**

### **Bibliografía**

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

- [Gay 95] V. GAY, P. LEYDEKKERS y 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, diciembre 1991.
- [Rumbaugh 91] J. RUMBAUGH y otros: Object oriented modelling and design, Prentice Hall, 1991.

## **SERIES DE RECOMENDACIONES DEL UIT-T**

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
<b>Serie X</b>	<b>Redes de datos y comunicación entre sistemas abiertos</b>
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes de programación