



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

X.902

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

(11/95)

**REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS**

PROCESAMIENTO DISTRIBUIDO ABIERTO

**TECNOLOGÍA DE LA INFORMACIÓN –
PROCESAMIENTO DISTRIBUIDO ABIERTO –
MODELO DE REFERENCIA: FUNDAMENTOS**

Recomendación UIT-T X.902

(Anteriormente «Recomendación del CCITT»)

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. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

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. El texto de la Recomendación UIT-T X.902 se aprobó el 21 de noviembre de 1995. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 10746-2.

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.

© UIT 1997

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.

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

(Febrero de 1994)

ORGANIZACIÓN DE LAS RECOMENDACIONES DE LA SERIE X

Dominio	Recomendaciones
REDES PÚBLICAS DE DATOS	
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
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
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 enunciados 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
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300-X.349
Sistemas móviles de transmisión de datos	X.350-X.369
Gestión	X.370-X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400-X.499
DIRECTORIO	X.500-X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600-X.649
Denominación, direccionamiento y registro	X.650-X.679
Notación de sintaxis abstracta uno	X.680-X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.700-X.799
SEGURIDAD	X.800-X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Cometimiento, concurrencia y recuperación	X.850-X.859
Tratamiento de transacciones	X.860-X.879
Operaciones a distancia	X.880-X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900-X.999

ÍNDICE

		<i>Página</i>
1	Alcance.....	1
2	Referencias normativas	1
	2.1 Recomendaciones Normas Internacionales idénticas.....	1
3	Definiciones	1
	3.1 Definiciones tomadas de otros documentos	1
	3.2 Definiciones generales	1
4	Abreviaturas	2
5	Categorías de conceptos	2
6	Conceptos básicos de interpretación	3
7	Conceptos básicos lingüísticos.....	3
8	Conceptos básicos de modelado.....	3
9	Conceptos de especificación	5
	9.1 composición	5
	9.2 objeto compuesto	6
	9.3 descomposición.....	6
10	Conceptos de organización.....	9
11	Propiedades de sistemas y objetos.....	10
	11.1 Transparencias	10
	11.2 Conceptos de políticas	11
	11.3 Propiedades temporales.....	12
12	Conceptos de denominación.....	12
13	Conceptos relativos al comportamiento	12
	13.1 Estructura de actividad.....	12
	13.2 Comportamiento contractual.....	13
	13.3 Causalidad.....	14
	13.4 Comportamientos de establecimiento	14
	13.5 Seguridad de funcionamiento.....	15
14	Conceptos de gestión.....	16
15	Enfoque de la conformidad atendiendo al ODP	16
	15.1 Conformidad con las normas ODP	16
	15.2 Pruebas y puntos de referencia.....	16
	15.3 Clases de puntos de referencia	16
	15.4 Cambio de configuración	17
	15.5 Proceso de prueba de conformidad	17
	15.6 Resultado de las pruebas.....	18
	15.7 Relación entre puntos de referencia	18

Resumen

Esta Recomendación | Norma Internacional contiene la definición de los conceptos y el marco analítico para la descripción normalizada de sistemas de procesamiento distribuido (arbitrarios). Presenta los principios de conformidad con normas de procesamiento distribuido abierto (ODP) y la manera de aplicarlos, sólo a un nivel de detalle suficiente para sustentar la Rec. X.903 | ISO/CEI 10746-2 y establecer los requisitos para las nuevas técnicas de especificación.

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 (RM-ODP, *reference model of open distributed processing*), Recs. UIT-T X.901 a X.904 | ISO/CEI 10746, 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 está constituido por:

- Rec. UIT-T X.901 | ISO/CEI 10746-1: **Visión de conjunto**: contiene una visión de conjunto de las motivaciones 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, que pueden incluir escritores de normas y arquitectos de sistemas ODP. Contiene también una agrupación en categorías de las áreas requeridas de normalización, expresadas en términos de los puntos de referencia para conformidad identificados en la Rec. UIT-T X.903 | ISO/CEI 10746-3. Esta parte no es normativa.
- 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). Introduce los principios de conformidad con las normas ODP y la forma en que deben aplicarse. 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. Esta parte es normativa.
- 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. Estas 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. Esta parte es normativa.
- Rec. UIT-T X.904 | ISO/CEI 10746-4: **Semántica arquitectural**: contiene una formalización de los conceptos de modelado ODP definidos en esta Recomendación | Norma Internacional (cláusulas 8 y 9). La formalización se consigue interpretando cada concepto en término de las construcciones de las diferentes técnicas de descripción formal normalizadas. Esta parte es normativa.

Esta Recomendación | Norma Internacional no contiene anexos.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

TECNOLOGÍA DE LA INFORMACIÓN – PROCESAMIENTO DISTRIBUIDO ABIERTO – MODELO DE REFERENCIA: FUNDAMENTOS

1 Alcance

Esta Recomendación UIT-T | Norma Internacional trata los conceptos necesarios para el modelado de sistemas ODP (véanse las cláusulas 5 a 14) y los principios de conformidad con los sistemas ODP (véase la cláusula 15).

Los conceptos definidos en las cláusulas 5 a 15 se utilizan en el modelo de referencia de procesamiento distribuido abierto como base para la definición de:

- a) la estructura de la familia de normas que deben respetar el modelo de referencia;
- b) la estructura de los sistemas distribuidos que pretenden el cumplimiento del modelo de referencia (la configuración de los sistemas);
- c) los conceptos necesarios para expresar la utilización combinada de las distintas normas aplicadas;
- d) los conceptos básicos que habrán de utilizarse en las especificaciones de los diversos componentes que constituyen el sistema distribuido abierto.

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 Internacionales 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.903 (1995) | ISO/CEI 10746-3:1996, *Tecnología de la información – Procesamiento distribuido abierto – Modelo de referencia: Arquitectura*.

3 Definiciones

A los efectos de esta Recomendación | Norma Internacional se aplican las siguientes definiciones.

3.1 Definiciones tomadas de otros documentos

En esta Recomendación | Norma Internacional no se han tomado definiciones de otras Recomendaciones | Normas Internacionales.

3.2 Definiciones generales

3.2.1 procesamiento distribuido: Procesamiento de información en el que componentes discretos pueden estar situados en diferentes lugares, y en el que la comunicación entre componentes puede sufrir retardos o fallos.

3.2.2 normas de procesamiento distribuido abierto (ODP): Este modelo de referencia y las normas que se ajustan al mismo, directa o indirectamente.

3.2.3 procesamiento distribuido abierto: Procesamiento distribuido destinado a cumplir las normas de procesamiento distribuido abierto.

3.2.4 sistema de procesamiento distribuido abierto: Sistema (véase 6.5) que cumple los requisitos de las normas de procesamiento distribuido abierto.

3.2.5 información: Toda clase de conocimiento, que pueda intercambiarse entre usuarios, sobre cosas, hechos, conceptos, etc. en un universo de discurso.

Aunque será necesario dar a la información una forma de representación para poder comunicarla, lo que interesa primordialmente es la interpretación (el significado) de esa representación.

3.2.6 datos: Las formas de representación de información que son tratadas por los sistemas de información y por los usuarios de estos sistemas.

3.2.7 punto de vista (sobre un sistema): Forma de abstracción obtenida utilizando un conjunto seleccionado de conceptos arquitecturales y reglas de estructuración, para destacar aspectos particulares de interés dentro de un sistema.

4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional se utilizan las siguientes abreviaturas:

ODP	Procesamiento distribuido abierto (<i>open distributed processing</i>)
OSI	Interconexión de sistemas abiertos (<i>open systems interconnection</i>)
PICS	Declaración de conformidad de implementación de protocolo (<i>protocol implementation conformance statement</i>)
PIXIT	Información suplementaria de implementación de protocolo para pruebas (<i>protocol implementation extra information for testing</i>)
RM-ODP	Modelo de referencia de procesamiento distribuido abierto (<i>reference model of open distributed processing</i>)
TP	Procesamiento de transacciones (<i>transaction processing</i>)

5 Categorías de conceptos

Los conceptos de modelado definidos en esta Recomendación | Norma Internacional se agrupan en las siguientes categorías:

- Conceptos básicos de interpretación:* conceptos para la interpretación de las construcciones de modelado de cualquier lenguaje de modelado ODP. Se describen en la cláusula 6.
- Conceptos básicos lingüísticos:* conceptos relacionados con lenguajes; la descripción de la gramática de cualquier lenguaje para la arquitectura ODP deberá basarse en estos conceptos. Se describen en la cláusula 7.
- Conceptos básicos de modelado:* conceptos para la construcción de la arquitectura ODP. Las construcciones de modelado de cualquier lenguaje deberán basarse en estos conceptos. Se describen en la cláusula 8.
- Conceptos de especificación:* conceptos relacionados con los requisitos de los lenguajes de especificación elegidos, utilizados en ODP. Estos conceptos no son inherentes a la distribución y a los sistemas distribuidos, sino que constituyen requisitos que habrán de considerarse en estos lenguajes de especificación. Estos conceptos se describen en la cláusula 9.
- Conceptos de estructuración:* conceptos que surgen cuando se consideran diferentes cuestiones en materia de distribución y en sistemas distribuidos. Puede que estén o no directamente soportados por lenguajes de especificación adecuados para tratar el área problemática. La especificación de objetos y funciones que soportan directamente estos conceptos se hará posible por la utilización de los lenguajes de especificación elegidos. Estos conceptos se describen en las cláusulas 10 a 14.
- Conceptos de conformidad:* conceptos necesarios para explicar las nociones de conformidad con las normas ODP y las de pruebas de conformidad. Estos conceptos se describen en la cláusula 15.

La Rec. UIT-T X.903 | ISO/CEI 10746-3 utiliza los conceptos definidos en la presente Recomendación | Norma Internacional para especificar las características que debe tener un procesamiento distribuido para que sea abierto. Está organizada como un conjunto de lenguajes de punto de vista. Cada lenguaje de punto de vista afina conceptos del conjunto definido en esta Recomendación | Norma Internacional. No es necesario que todos los lenguajes de punto de vista adopten la misma notación. Para reflejar los requisitos del punto de vista pueden elegirse, como notaciones apropiadas, notaciones que sean diferentes. Estas notaciones pueden ser la notación natural, formal, textual o gráfica. Sin embargo, será necesario establecer correspondencias entre los diversos lenguajes para asegurar una coherencia global.

6 Conceptos básicos de interpretación

Aunque gran parte de la arquitectura ODP se ocupa de la definición de construcciones formales, hay que describir la semántica del modelo arquitectural y de los lenguajes de modelado utilizados. Estos conceptos son, esencialmente, metaconceptos, es decir, conceptos que se aplican por lo general a cualquier forma de actividad de modelado. No se pretende que estos conceptos sean definidos formalmente, ni que sirvan de base para la definición formal de otros conceptos.

Toda actividad de modelado identifica:

- a) elementos del universo de discurso;
- b) uno o más niveles de abstracción pertinentes.

Los elementos del universo de discurso son entidades y proposiciones.

6.1 entidad: Toda cosa de interés, sea concreta o abstracta. Si bien, en general, la palabra entidad puede utilizarse para designar cualquier cosa, en el contexto del modelado se emplea exclusivamente para designar cosas en el universo de discurso que se está modelando.

6.2 proposición: Hecho o estado de cosas observable que comprende unas o más entidades, con respecto a la cual se puede afirmar su certeza, o negar que sea válida con respecto a esas entidades.

6.3 abstracción: El proceso de suprimir detalles no significativos para el establecimiento de un modelo simplificado, o el resultado de ese proceso.

6.4 atomicidad: Una entidad es atómica en un nivel dado de abstracción si no puede dividirse en ese nivel de abstracción.

La fijación de un nivel de abstracción puede implicar la identificación de los elementos que son atómicos.

6.5 sistema: Algo que despierta interés en su conjunto, atendiendo a las partes de que se compone. Por consiguiente, se puede hacer referencia a un sistema como una entidad. Un componente de un sistema puede ser a su vez un sistema, en cuyo caso se puede decir que es un subsistema.

NOTA – A los fines del modelado, el concepto de sistema debe entenderse en su sentido general, de lo que teóricamente es un sistema. El término «sistema» puede referirse a un sistema de procesamiento de la información, pero puede utilizarse también en una acepción más general.

6.6 arquitectura (de un sistema): Conjunto de reglas para definir la estructura de un sistema y las relaciones entre las partes.

7 Conceptos básicos lingüísticos

Un lenguaje de modelado para la arquitectura ODP, cualesquiera que sean sus conceptos o semántica, se expresará en alguna sintaxis, que puede incluir texto lineal o convenios gráficos. Se supone que todo lenguaje adecuado tendrá una gramática que definirá el conjunto válido de símbolos y construcciones lingüísticas bien formadas del lenguaje. Los siguientes conceptos proporcionan un marco común para relacionar la sintaxis de cualquier lenguaje utilizado para la arquitectura ODP con los conceptos de interpretación.

7.1 término: Construcción lingüística que puede utilizarse para hacer referencia a una entidad.

La referencia puede ser a cualquier clase de entidad, incluido un modelo de una entidad u otra construcción lingüística.

7.2 oración: Una construcción lingüística que contiene uno o más términos y predicados; una oración puede utilizarse para expresar una proposición sobre las entidades a que se refiere el término.

Puede considerarse que un predicado en una oración se refiere a una relación entre las entidades a que hacen referencia los términos enlazados.

8 Conceptos básicos de modelado

La interpretación detallada de los conceptos definidos en esta cláusula dependerá del lenguaje de especificación utilizado, pero estos enunciados generales de conceptos se hacen de una forma independiente del lenguaje para permitir la interrelación de enunciados expresados en lenguajes diferentes.

Los conceptos básicos se refieren a la existencia y actividad de algo: la expresión de qué (qué cosa) existe, dónde está, y qué hace.

8.1 objeto: Un modelo de una entidad. Un objeto se caracteriza por su comportamiento (véase 8.6) y, sobre una base dual, por su estado (véase 8.7). Un objeto es distinto de cualquier otro objeto. Un objeto está encapsulado, es decir, todo cambio en su estado sólo puede producirse como resultado de una acción interna o como resultado de una interacción (véase 8.3) con su entorno (véase 8.2).

Un objeto interactúa con su entorno en sus puntos de interacción (véase 8.11).

En función del punto de vista, se puede dar más peso al comportamiento o al estado. Cuando se da más peso al comportamiento, se dice informalmente de un objeto que realiza funciones y ofrece servicios (de un objeto que pone a disposición una función se dice que ofrece un servicio). A los fines del modelado, estas funciones y servicios se especifican en términos del comportamiento del objeto y de sus interfaces (véase 8.4). Un objeto puede realizar más de una función. Una función puede ser realizada por la cooperación de varios objetos.

NOTAS

1 Los conceptos de servicio y función se utilizan informalmente para expresar la finalidad de una pieza de normalización. En la familia ODP de normas, función y servicio se expresan formalmente en términos de la especificación del comportamiento de objetos y de los interfaces de éstos. Un «servicio» es una abstracción particular del comportamiento que expresa las garantías ofrecidas por un proveedor de servicio.

2 La expresión «utilización de una función» es una abreviación para la interacción con un objeto que realiza la función.

8.2 entorno (de un objeto): Todo lo que forma parte del modelo y no forma parte de ese objeto.

NOTA – En muchos lenguajes de especificación puede considerarse que el entorno incluye por lo menos un objeto capaz de participar, sin restricción alguna, en todas las interacciones posibles (véase 8.3); este objeto representa el proceso de observación.

8.3 acción: Algo que se realiza (o algo que sucede, u ocurre).

Toda acción de interés para fines de modelado está asociada por lo menos con un objeto.

El conjunto de acciones asociadas con un objeto se divide en **acciones internas** e **interacciones**. Una acción interna siempre se realiza sin participación del entorno del objeto. Una interacción se realiza con la participación del entorno del objeto.

NOTAS

1 «Acción» significa «realización de la acción» (u «ocurrencia de la acción»). En función del contexto, una especificación puede expresar que una acción se ha realizado (ha ocurrido), está realizándose (está ocurriendo) o puede realizarse (puede ocurrir).

2 La granularidad de las acciones se determina en el diseño. Una acción no tiene que ser instantánea. Las acciones pueden superponerse en el tiempo.

3 Las acciones pueden ser etiquetadas en base a relaciones de causa a efecto entre los objetos participantes. Los conceptos correspondientes se examinan en 13.3.

4 Un objeto puede interactuar consigo mismo, en cuyo caso se considera que desempeña dos papeles en la interacción; en este contexto, puede considerarse que el objeto forma parte de su propio entorno.

5 La intervención del entorno implica la posibilidad de observar lo que sucede en su interior. Las interacciones son, por tanto, observables; en cambio, las acciones internas no son observables debido a la encapsulación de los objetos.

8.4 interfaz: Una abstracción del comportamiento de un objeto que consiste en un subconjunto de las interacciones de ese objeto, y un conjunto de restricciones sobre el momento en que deben ocurrir dichas interacciones.

Cada interacción de un objeto pertenece a una interfaz única. Por tanto, las interfaces de un objeto forman una partición de las interacciones de ese objeto.

NOTAS

1 Una interfaz constituye la parte del comportamiento de un objeto que se obtiene considerando solamente las interacciones de esa interfaz y ocultando todas las otras interacciones. La ocultación de las interacciones de otras interfaces introduce generalmente un no determinismo en lo que respecta a la interfaz considerada.

2 La frase «una interfaz entre objetos» se utiliza para designar la vinculación (véase 13.4.2) entre interfaces de los objetos que intervienen.

8.5 actividad: Gráfico acíclico, dirigido, y de una sola cabeza, que representa acciones, en el que la ocurrencia de cada acción en el gráfico se hace posible por la ocurrencia de todas las acciones que le preceden inmediatamente (por ejemplo, de todas las acciones adyacentes que están más próximas a la cabeza).

8.6 comportamiento (de un objeto): Una colección de acciones con un conjunto de restricciones sobre el momento en que aquéllas pueden ocurrir.

El lenguaje de especificación que se está utilizando determina las restricciones que pueden expresarse. Las restricciones pueden incluir, por ejemplo, las de secuenciabilidad, no determinismo, concurrencia, o tiempo real.

Un comportamiento puede incluir acciones internas.

Las acciones que efectivamente pueden realizarse están restringidas por el entorno en que se coloca el objeto.

NOTAS

1 La composición (véase 9.1) de una colección de objetos da implícitamente un objeto equivalente que representa la composición. Se suele hacer referencia al comportamiento de este objeto, sencillamente, como el comportamiento de la colección de objetos.

2 Acción y actividad son casos degenerados de comportamiento.

3 En general, varias secuencias de interacciones son consecuentes con un comportamiento dado.

8.7 estado (de un objeto): En un instante dado, la condición de un objeto que determina el conjunto de todas las secuencias de acciones en que el objeto puede participar.

Puesto que, en general, el comportamiento incluye muchas posibles series de acciones en que el objeto puede participar, el conocimiento del estado no permite necesariamente la predicción de la secuencia de acciones que realmente se producirá.

Los cambios de estado se efectúan por acciones; en consecuencia, un estado es parcialmente determinado por las acciones precedentes en que participó el objeto.

Puesto que un objeto está encapsulado, su estado no se puede cambiar directamente desde el entorno, sino sólo indirectamente como resultado de las interacciones en el objeto participa.

8.8 comunicación: El transporte de información entre dos o más objetos como resultado de una o más interacciones, que posiblemente comprendan algunos objetos intermedios.

NOTAS

1 Las comunicaciones pueden ser etiquetadas en base a una relación de causa a efecto entre los objetos participantes. Los conceptos correspondientes se examinan en 13.3.

2 Toda interacción es un ejemplar (suele decirse una instancia) de una comunicación.

8.9 ubicación en el espacio: Un intervalo de tamaño arbitrario, en el espacio, en el que puede realizarse una acción.

8.10 ubicación en el tiempo: Un intervalo de tamaño arbitrario, en el tiempo, en el que puede realizarse una acción.

NOTAS

1 La magnitud del intervalo en el tiempo o en el espacio se elige de modo que refleje los requisitos de una determinada tarea de especificación y las propiedades de un determinado lenguaje de especificación. Una ubicación individual en una especificación puede estar dividida en el tiempo, en el espacio, o en el tiempo y en el espacio, en otra especificación. En una especificación dada, una ubicación en el espacio o en el tiempo se define con relación a algún sistema adecuado de coordenadas.

2 Por extensión, la ubicación de un objeto es la unión de las ubicaciones de las acciones en que el objeto puede participar.

8.11 punto de interacción: Ubicación en que existe un conjunto de interfaces.

En cualquier ubicación dada en el tiempo, un punto de interacción está asociado con una ubicación en el espacio, dentro de la especificidad permitida por el lenguaje de especificación que se está utilizando. Pueden existir varios puntos de interacción en una misma ubicación. Un punto de interacción puede ser móvil.

9 Conceptos de especificación

9.1 composición

a) (de objetos) – Combinación de dos o más objetos que da un nuevo objeto, en un nivel de abstracción diferente. Las características del nuevo objeto están determinadas por los objetos combinados y la manera de combinarlos. El comportamiento de un objeto compuesto es la composición correspondiente de los comportamientos de los objetos componentes.

b) (de comportamientos) – Combinación de dos o más comportamientos que da un nuevo comportamiento. Las características del comportamiento resultante están determinadas por los comportamientos combinados y la manera de combinarlos.

NOTAS

1 Son ejemplos de técnicas de combinación las siguientes: composición secuencial, composición concurrente, entrelazado, elección, y ocultación de acciones. Estas definiciones generales se utilizarán siempre en un sentido particular que identifica un determinado medio de combinación.

2 En algunos casos, la composición de comportamientos puede dar un comportamiento degenerado, por ejemplo una situación de bloqueo, debido a constricciones de los comportamientos componentes.

9.2 objeto compuesto: Objeto expresado como una composición.

9.3 descomposición

- a) (de un objeto) – Especificación de un objeto dado como una composición.
- b) (de un comportamiento) – Especificación de un comportamiento dado como una composición.

Composición y descomposición son términos duales y actividades de especificación duales.

9.4 compatibilidad en comportamiento: Un objeto es compatible en comportamiento con un segundo objeto respecto a un conjunto de criterios (véanse las Notas) si el primer objeto puede reemplazar al segundo objeto sin que el entorno pueda percibir la diferencia en el comportamiento de los objetos sobre la base del conjunto de criterios.

Típicamente, los criterios imponen constricciones al comportamiento permitido del entorno. Si, según los criterios, el entorno se comporta como un probador para el objeto original, es decir, el entorno define el comportamiento más pequeño que no constriñe el comportamiento del objeto original, la relación resultante de compatibilidad en comportamiento se denomina extensión.

Los criterios pueden permitir que el objeto reemplazante sea derivado por modificación de un objeto que en otro caso sería incompatible para que, de ese modo, el reemplazo sea aceptable. Un ejemplo de tal modificación pudiera ser la ocultación de parámetros adicionales en ciertas interacciones. De esta forma, puede hacerse que una interacción del nuevo objeto parezca una interacción del objeto original. En esos casos, se dice que la compatibilidad en comportamiento es una **compatibilidad en comportamiento coercida**. Si no es necesaria una modificación se dice que la compatibilidad en comportamiento es una **compatibilidad en comportamiento natural**.

El concepto de compatibilidad en comportamiento definido anteriormente con relación a objetos se aplica asimismo a la compatibilidad en comportamiento de plantillas y de tipos de plantillas.

La compatibilidad en comportamiento es reflexiva, pero no necesariamente simétrica o transitiva (puede ser simétrica, o transitiva, o simétrica y transitiva).

NOTAS

- 1 El conjunto de criterios depende del lenguaje utilizado y de la teoría de prueba aplicada.
- 2 La compatibilidad en comportamiento (con respecto a un conjunto de criterios) puede definirse atendiendo a la plantilla (véase 9.11) y a los tipos de plantilla (véase 9.19), de la manera siguiente:
 - a) si S y T son plantillas de objeto, se dice que S es compatible en comportamiento con T si, y solamente si, cualquier instanciación de S es compatible en comportamiento con alguna instanciación de T (véase 9.13);
 - b) si U y V son tipos de plantilla de objeto, se dice que U y V son compatibles en comportamiento si sus correspondientes plantillas son *compatibles en comportamiento*.

9.5 refinamiento: Proceso de transformación de una especificación en una especificación más detallada. Se puede hacer referencia a la nueva especificación como un refinamiento de la especificación original. Típicamente, las especificaciones y sus refinamientos no coexisten en la misma descripción de sistema. Lo que ha de entenderse precisamente por una especificación más detallada dependerá del lenguaje de especificación utilizado.

Para cada significado de compatibilidad en comportamiento determinado por algún conjunto de criterios (véase 9.4), una técnica de especificación permitirá la definición de una relación de refinamiento. Si la plantilla X es un refinamiento de la plantilla Y, será posible reemplazar un objeto instanciado a partir de Y por uno instanciado a partir de X en el conjunto de entornos determinado por la definición seleccionada de compatibilidad en comportamiento. Las relaciones de refinamiento no son necesariamente o bien simétricas, o bien transitivas.

9.6 rastro: Un registro de las interacciones de un objeto, desde su estado inicial hasta algún otro estado.

Un rastro de un objeto es por tanto una secuencia finita de interacciones. El comportamiento determina unívocamente el conjunto de todos los rastros posibles, pero a la inversa. Un rastro no contiene un registro de las interacciones de un objeto.

9.7 tipo (de una <X>): Un predicado que caracteriza una colección de <X>s. Una <X> es del tipo, o satisface el tipo, en cuestión, si el predicado es válido para esa <X>. Una especificación define cuáles de los términos que ella utiliza tienen tipos, es decir, son <X>s. En RM-ODP, se necesitan tipos para, por lo menos, objetos, interfaces y acciones.

La noción de tipo permite clasificar las entidades en categorías, algunas de las cuales pueden ser de interés para el especificador (véase el concepto de clase en 9.8).

9.8 clase (de <X>s): El conjunto de todos las <X>s que satisfacen un tipo (véase 9.7). Los elementos del conjunto se llaman miembros de la clase.

NOTAS

- 1 Una clase puede no tener miembros.
- 2 El hecho de que el tamaño del conjunto varíe o no en función del tiempo depende de la definición del tipo.

9.9 subtipo/supertipo: Un tipo A es un subtipo de B, y B es un supertipo de A, si toda <X> que satisface a A satisface también a B.

Las relaciones de subtipo y supertipo son reflexivas, transitivas antisimétricas.

9.10 subclase/superclase: Una clase A es una subclase de otra clase B, y B es una superclase de A, precisamente cuando el tipo asociado con A es un subtipo del tipo asociado con B.

NOTA – Una subclase es por definición un subconjunto de cualquiera de sus superclases.

9.11 plantilla <X>: Especificación de las características comunes de una colección de <X>s suficientemente detallada para que, mediante ella, se pueda instanciar una <X>. <X> puede ser cualquier cosa que tenga un tipo (véase 9.7).

Una plantilla <X> es una abstracción de una colección de <X>s.

Una plantilla puede especificar parámetros que serán vinculados en el momento de la instanciación.

La definición que aquí se da es genérica; la forma precisa de una plantilla dependerá de la técnica de especificación utilizada. Los tipos de parámetros (donde sean aplicables) también dependerán de la técnica de especificación utilizada.

Las plantillas pueden combinarse mediante cálculo. La forma precisa de la combinación de las plantillas dependerá del lenguaje de especificación utilizado.

9.12 firma de interfaz: Conjunto de plantillas de acciones asociado con las interacciones de una interfaz.

Un objeto puede tener muchas interfaces con la misma firma.

9.13 instanciación (de una plantilla <X>): Una <X> producida a partir de una plantilla <X> dada y de otra información necesaria. Esta <X> tiene las características especificadas en la plantilla <X>. <X> puede ser cualquier cosa que tenga un tipo (véase 9.7).

La definición que aquí se da es genérica; la manera de instanciar una plantilla <X> depende del lenguaje de especificación utilizado. La instanciación de una plantilla <X> puede implicar la actualización de parámetros, lo que a su vez puede implicar la instanciación de otras plantillas <X> o la vinculación de interfaces existentes (véase 12.4).

NOTAS

- 1 La instanciación de una plantilla de acción tiene por resultado, justamente, la realización de una acción.
- 2 Si <X> es un objeto, su instanciación produce un objeto en su estado inicial. Un objeto puede participar en interacciones inmediatamente después de su instanciación.
- 3 Instanciaciones efectuadas a partir de plantillas diferentes pueden satisfacer el mismo tipo. Instanciaciones efectuadas a partir de la misma plantilla pueden satisfacer tipos diferentes.

9.14 rol (sinónimo: papel): Identificador para un comportamiento, que puede aparecer como un parámetro en una plantilla para un objeto compuesto, y que se asocia a uno de los objetos componentes del objeto compuesto.

La especificación de una plantilla como una composición de roles permite explicar el proceso de instanciación como la asociación de un componente específico del objeto compuesto resultante, con cada rol. La asociación de un objeto componente con un rol puede ser consecuencia de la actualización de un parámetro.

9.15 creación (de una <X>): Instanciación de una <X> cuando se obtiene por una acción de objetos en el modelo. <X> es cualquier cosa que pueda ser instanciada; en particular, puede ser un objeto o una interfaz.

Si <X> es una interfaz se crea, o bien con ocasión de la creación de un objeto dado, o bien como una interfaz adicional al objeto creador. Por consiguiente, toda interfaz tiene que formar parte de un objeto.

9.16 introducción (de una <X>): Instanciación de una <X> cuando no se obtiene por una acción de objetos en el modelo.

NOTAS

- 1 Una <X> puede ser instanciada sea por creación, sea por introducción, pero no de ambos modos.
- 2 La introducción no se aplica a interfaces y acciones, porque tanto las unas como las otras son siempre soportadas por objetos.

9.17 supresión (de una <X>): Acción de destruir una <X> instanciada. <X> es cualquier cosa que pueda ser instanciada; en particular, puede ser un objeto o una interfaz.

Si <X> es una interfaz, sólo puede ser suprimida por el objeto a que está asociada.

NOTA – La supresión de una acción no tiene sentido: una acción sólo acontece (es realizada).

9.18 instancia (de un tipo): Una <X> que satisface el tipo.

9.19 tipo de plantilla (de una <X>): Un predicado definido en una plantilla que es válido para todas las instanciaciones de la plantilla y que expresa los requisitos que se tiene previsto serán cumplidos por las instanciaciones de la plantilla.

La relación subtipo/supertipo de plantilla de objeto no coincide necesariamente con la compatibilidad en comportamiento. Las instancias de un tipo de plantilla no tienen que ser compatibles en comportamiento con instanciaciones de la plantilla asociada. Coinciden, en efecto, si se cumplen las dos condiciones siguientes:

- si se considera una relación de compatibilidad en comportamiento transitiva, y
- si los subtipos de plantilla son compatibles en comportamiento con sus supertipos de plantilla.

NOTAS

- Este concepto encierra la noción de la posibilidad de sustitución por diseño.
- La forma del predicado que expresa el tipo de plantilla depende del lenguaje de especificación utilizado.
- Como una abreviación, habrá de entenderse, por definición, que «instancias de una plantilla T» son «instancias del tipo de plantilla asociado con la plantilla T».
- La Figura 1 ilustra las relaciones entre algunos de los conceptos: tipo de plantilla, clase de plantilla, etc. El conjunto de instancias de **t** contiene el conjunto de instanciaciones de **t** y los conjuntos de todas las instanciaciones de subtipos de **t**. Los conjuntos de instanciaciones de plantillas diferentes son siempre disjuntos.

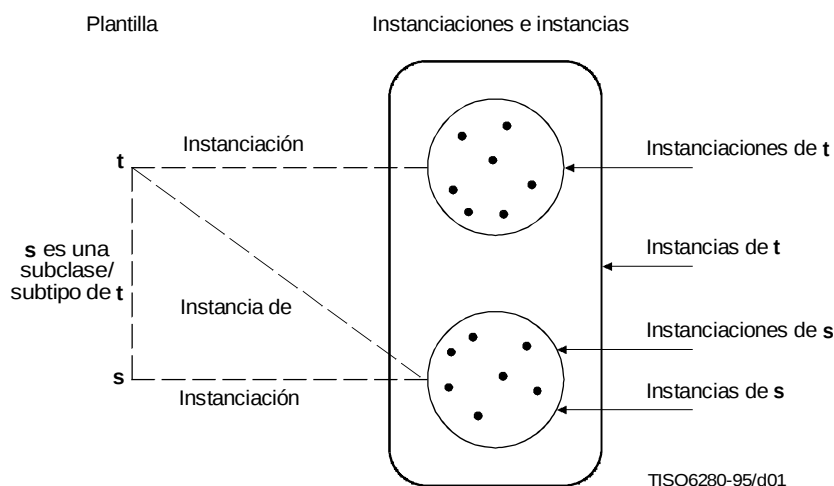


Figura 1 – Relación entre plantillas, instanciaciones e instancias

9.20 clase (de) plantilla (de una <X>): Conjunto de todas las <X> que satisfacen un tipo de plantilla <X>, es decir, el conjunto de <X>s que son instancias de la plantilla <X>. <X> puede ser cualquier cosa que tenga un tipo (véase 9.7).

Cada plantilla define una sola clase de plantilla, por lo que las instancias de plantillas podemos llamarlas instancias de la clase plantilla.

La noción de clase se utiliza para hacer referencia a una clasificación general de <X>s. Clase plantilla es una noción más restrictiva en la que los miembros de una clase plantilla están limitados a los instanciados a partir de la plantilla (o de cualquiera de sus subtipos), es decir, a las <X> que satisfacen el tipo de plantilla <X>.

NOTA – Dado un tipo de plantilla, podemos abreviar enunciados de la forma «la clase plantilla asociada con la plantilla A es una subclase de la clase plantilla asociada con la plantilla B» expresándolos así: «plantilla A es una subclase de plantilla B» o «plantilla A es un subtipo de plantilla B».

9.21 clase derivada/clase de base: Si una plantilla A es una modificación incremental de una plantilla B, la clase plantilla CA de instancias de A es una clase derivada de la clase plantilla CB de instancias de B, y la CB es una clase de base de CA.

Los criterios para considerar que un cambio cualquiera es una modificación incremental dependerán de conceptos sobre la métrica y de convenios que están fuera del ámbito de esta Recomendación | Norma Internacional. Si los criterios lo permiten, una clase derivada puede tener varias clases de base.

La modificación incremental relativa a las plantillas asegurará que una autorreferencia o recursión en la plantilla de la clase de base se torna en una autorreferencia o recursión en la plantilla de la clase derivada.

La modificación incremental puede comprender, en general, la adición o alteración de propiedades de la plantilla de base para obtener la plantilla derivada.

Las clases pueden organizarse en una jerarquía de herencia de acuerdo con relaciones de clase derivada/clase de base. Esta es la interpretación de herencia en el modelo de referencia ODP. Si las clases pueden tener varias clases de base, se dice que la herencia es múltiple. Si los criterios prohíben la supresión de propiedades en la clase de base, se dice que la herencia es estricta.

Es posible que una clase sea una subclase de una segunda clase sin que sea una clase derivada, y que sea una clase derivada sin que sea una subclase. La jerarquía de herencia (donde los arcos representan la relación de clase derivada) y la jerarquía de tipo (donde los arcos representan la relación de subtipo o de subclase) son por tanto lógicamente distintas, aunque pueden coincidir total o parcialmente.

9.22 invariante: Predicado que la especificación requiere que sea verdadero durante todo el tiempo de vida de un conjunto de objetos.

9.23 precondition: Predicado que una especificación requiere que sea verdadero para que se realice una acción.

9.24 postcondición: Predicado que una especificación requiere que sea verdadero inmediatamente después de realizada una acción.

10 Conceptos de organización

10.1 grupo <X>: Conjunto de objetos con una relación particular de caracterización <X>. La relación <X> caracteriza sea la relación estructural entre objetos, sea un comportamiento común previsto de los objetos.

NOTA – Son ejemplos de grupos especializados:

- a) *Grupo direccionado:* conjunto de objetos direccionados de la misma manera.
- b) *Grupo en base a los fallos (faltas, o averías):* conjunto de objetos que tienen la característica común de depender del mismo fallo (falta, o avería). Por ejemplo, puede suponerse que si un computador falla, todos los objetos que ejecutan acciones sobre ese computador fallarán también.
- c) *Grupo comunicante:* un conjunto de objetos en el que todos los objetos participan en la misma secuencia de interacciones con su entorno.
- d) *Grupo de replicación tolerante a los fallos (faltas, averías):* un grupo comunicante que tiene por finalidad proporcionar cierto nivel de tolerancia a algunos fallos.

10.2 configuración (de objetos): Una configuración de objetos capaces de interactuar en interfaces. Una configuración determina el conjunto de objetos que participan en cada interacción.

La especificación de una configuración puede ser estática o basarse en la operación de mecanismos dinámicos que cambian la configuración, como son la vinculación y la desvinculación (véase 13.4).

NOTA – Una configuración puede expresarse en términos de los conceptos de composición concurrente. El proceso de composición genera un objeto equivalente para la configuración, a un nivel diferente de abstracción.

10.3 dominio <X>: Conjunto de objetos, cada uno de los cuales se relaciona, por una relación caracterizante <X>, con un objeto controlador.

Todo dominio tiene asociado un objeto controlador.

El objeto controlador puede determinar las identidades de la colección de objetos que constituyen el dominio asociado. El objeto controlador puede comunicar dinámicamente con un objeto controlado, o puede considerarse que ya ha comunicado en una época (véase 10.5) anterior del objeto controlador. Generalmente, el objeto controlador no es miembro del dominio asociado.

NOTAS

- 1 En términos de empresa, diversas políticas pueden ser administradas por el objeto controlador en todo el dominio.
- 2 Los dominios pueden ser disjuntos o superpuestos.

- 3 Por definición, un dominio es un grupo, pero el enunciado inverso no es válido.
- 4 Los siguientes son ejemplos de dominios especializados:

Dominio	Clase miembro	Relación	Clase controladora
Dominio de seguridad	objeto de procesamiento	sujeto a la política establecida por	objeto autoridad de seguridad
Dominio de gestión	objeto gestionado	sujeto a la política establecida por	objeto dominio de gestión
Dominio de direccionamiento	objeto direccionado	dirección atribuida por	objeto autoridad de direccionamiento
Dominio de denominación	objeto denominado	nombre atribuido por	objeto autoridad de denominación

10.4 subdominio: Un dominio que es un subconjunto de un dominio dado.

10.5 época: Un periodo de tiempo durante el cual un objeto tiene un comportamiento particular. Todo objeto está en una sola época en cada momento, pero los objetos interactuantes pueden estar en épocas diferentes en el tiempo de la interacción.

Un cambio de época puede ser asociado con un cambio en el tipo del objeto, a fin de soportar la evolución de tipo. Por otra parte, un cambio de época puede ser asociado con una fase del comportamiento de un objeto de tipo constante.

Para que un sistema distribuido funcione correctamente, los objetos que constituyen su configuración tienen que ser coherentes. Por tanto, cuando la totalidad del sistema evoluciona a través de una serie de épocas, los objetos individuales que interactúan nunca estarán en épocas en las cuales sus comportamientos sean lo suficientemente diferentes para que su composición concurrente conduzca a un fallo. Este concepto soportará la formalización de conceptos de versión y extensibilidad.

NOTA – Un lenguaje de especificación puede tener que expresar:

- la manera de etiquetar las épocas;
- la secuencia de épocas, y si todos los objetos deben o no pasar a través de todos los miembros de la secuencia;
- las reglas para derivar la época de una composición a partir de las épocas de sus objetos, en particular para configuraciones y sistemas completos;
- si la identidad de la época de un objeto forma o no necesariamente parte del estado de ese objeto;
- si los objetos pueden o no negociar sobre la base de sus identidades de época actuales;
- la relación de la época con los conceptos de tiempo (u hora) local y global.

10.6 punto de referencia: Punto de interacción definido en una arquitectura para su selección como un punto de conformidad en una especificación que cumple con esa arquitectura.

En ODP se identifican clases de punto referencia significativas; en la cláusula 15 se da información detallada de estas clases y de la relación del modelado con la conformidad.

10.7 punto de conformidad: Punto de referencia en el que puede observarse el comportamiento para fines de pruebas de conformidad.

11 Propiedades de sistemas y objetos

Esta cláusula describe las propiedades aplicables a un sistema ODP o una parte de un sistema ODP.

11.1 Transparencias

11.1.1 transparencia de distribución: La propiedad de ocultar a un determinado usuario el comportamiento potencial de algunas partes de un sistema distribuido.

NOTA – Los usuarios pueden ser, por ejemplo, usuarios finales, desarrolladores de aplicaciones e implementadores de funciones.

11.2 Conceptos de políticas

11.2.1 contrato: Acuerdo que rige una parte del comportamiento colectivo de un conjunto de objetos. Un contrato especifica obligaciones, permisos y prohibiciones relativas a los objetos que intervienen.

La especificación de un contrato puede incluir:

- a) una especificación de los diferentes roles (o papeles) que pueden desempeñar los objetos implicados en el contrato, y las interfaces asociadas con los roles;
- b) atributos de calidad de servicio (véase 11.2.2);
- c) indicaciones de duración o periodos de validez;
- d) indicaciones de comportamientos que invalidan el contrato;
- e) condiciones de vitalidad y seguridad.

NOTAS

1 Los objetos en un contrato no tienen que estar relacionados jerárquicamente, pero pueden encontrarse en una relación de par a par. Los requisitos de un contrato no tienen que aplicarse de la misma manera a todos los objetos de interés.

2 Un contrato puede aplicarse en un determinado punto de referencia de un sistema. En ese caso especifica el comportamiento que cabe prever en ese punto de referencia.

3 Una plantilla de objeto proporciona un ejemplo simple de un contrato. Una plantilla de objeto especifica el comportamiento común a una colección de objetos. Como tal, especifica lo que el entorno de cualquiera de esos objetos puede suponer sobre su comportamiento. Obsérvese que, en el caso de especificaciones parciales, una plantilla de objeto deja sin especificar el comportamiento de un objeto en ciertas circunstancias del entorno (por ejemplo, en caso de interacciones particulares); el contrato sólo abarca el comportamiento especificado.

11.2.2 calidad de servicio: Un conjunto de requisitos de calidad que debe cumplir el comportamiento colectivo de uno o más objetos.

En cuanto a la calidad de servicio puede proceder de dos maneras: se puede especificar en un contrato, o medirla e informar sobre ella después del evento.

La calidad de servicio puede ser parametrizada.

NOTA – La calidad de servicio comprende características tales como la velocidad de transferencia de información, efectos de los errores no detectados (errores «latentes»), la probabilidad de corte de la comunicación, la probabilidad de fallo de sistema, la probabilidad de fallo de almacenamiento, etc.

11.2.3 contrato de entorno: Contrato entre un objeto y su entorno, incluidas constricciones de calidad de servicio y constricciones de utilización y gestión.

Las constricciones de calidad de servicio incluyen:

- constricciones temporales (por ejemplo, límites de tiempo);
- constricciones de volumen (por ejemplo, caudal);
- constricciones de seguridad de funcionamiento, que abarca aspectos tales como la disponibilidad, fiabilidad, mantenibilidad, securización, y seguridad (por ejemplo, tiempo medio entre fallos).

Las constricciones de utilización y gestión incluyen:

- constricciones de ubicación (es decir, ubicaciones seleccionadas en el espacio y en el tiempo);
- constricciones de transparencia de distribución (es decir, transparencias de distribución seleccionadas).

Las constricciones de calidad de servicio pueden implicar constricciones de utilización y gestión. Por ejemplo, algunas constricciones de calidad de servicio (como las de disponibilidad) se satisfacen proporcionando una o más transparencias de distribución (por ejemplo, la transparencia de replicación).

Las constricciones de entorno pueden describir:

- requisitos impuestos al entorno de un objeto para que el comportamiento del objeto sea correcto;
- constricciones sobre el comportamiento de un objeto en un entorno correcto.

11.2.4 obligación: Prescripción que establece que se requiere un determinado comportamiento. Una obligación se cumple cuando ocurre el comportamiento prescrito.

11.2.5 permiso: Prescripción que establece que la ocurrencia de un determinado comportamiento está autorizada. Un permiso es equivalente a que no exista la obligación de que el comportamiento en cuestión no ocurra.

11.2.6 prohibición: Prescripción que establece que un determinado comportamiento no deberá ocurrir. Una prohibición es equivalente a que exista la obligación de que el comportamiento en cuestión no ocurra.

11.2.7 política: Un conjunto de reglas relacionadas con una finalidad determinada.

NOTA – No toda política es una restricción. Algunas políticas representan el otorgamiento de una facultad («facultación»).

11.3 Propiedades temporales

11.3.1 persistencia: La propiedad que tiene un objeto de continuar existiendo después de producirse cambios de contexto contractual (véase 13.2.3) o de época.

11.3.2 isocronicidad: Una secuencia de acciones es isócrona si los intervalos de tiempo ocupados por cada uno de los pares de acciones adyacentes en la secuencia son, cada uno de ellos, únicos y del mismo tamaño.

12 Conceptos de denominación

12.1 nombre: Término que, en un determinado contexto de denominación, hace referencia a una entidad.

12.2 identificador: Nombre inequívoco, en un determinado contexto de denominación.

12.3 espacio de nombres: Un conjunto de términos que pueden usarse como nombres.

12.4 contexto de denominación: Relación entre un conjunto de nombres y un conjunto de entidades. El conjunto de nombres pertenece a un solo espacio de nombres.

12.5 acción de denominación: Acción que asocia un término de un espacio de nombres con una entidad dada.

Todas las acciones de denominación son relativas a un contexto de denominación.

12.6 dominio de denominación: Un subconjunto de un contexto de denominación, tal que todas las acciones de denominación las realiza el objeto controlador del dominio (el objeto autoridad de nombres).

NOTA – «Dominio de denominación» es una instancia del concepto dominio <X> (véase 10.3).

12.7 gráfico de denominación: Un gráfico dirigido en el que cada vértice representa un contexto de denominación, y cada borde una asociación entre:

- un nombre que aparece en el contexto de denominación fuente; y
- el contexto de denominación de destino.

NOTA – La existencia de un borde entre dos contextos de denominación en un gráfico de denominación significa que el contexto de denominación de destino puede ser alcanzado (identificado) desde el contexto de denominación fuente.

12.8 resolución de nombre: Proceso por el cual, dado un nombre inicial y un contexto de denominación inicial, puede encontrarse una asociación entre un nombre y la entidad designada por el nombre inicial.

NOTA – El proceso de resolución de nombre no proporciona necesariamente información suficiente para interactuar con la entidad designada.

13 Conceptos relativos al comportamiento

13.1 Estructura de actividad

13.1.1 cadena (de acciones): Secuencia de acciones dentro de una actividad en la que, para cada par adyacente de acciones, la realización (u ocurrencia) de la primera acción es necesaria para que se realice (u ocurra) la segunda.

13.1.2 hilo: Cadena de acciones en la que por lo menos un objeto participa en todas las acciones de la cadena.

Un objeto puede tener asociado un solo hilo, o muchos hilos al mismo tiempo.

13.1.3 acción de unión: Acción compartida entre dos o más cadenas como resultado de la cual se forma una sola cadena.

13.1.4 acción de división: Acción que habilita dos o más cadenas.

Hay dos casos de acción de división, según que se requiera o no que las dos cadenas habilitadas se reúnan ulteriormente.

13.1.5 acción de ramificación temporal: Acción de división en la que las cadenas habilitadas tienen que volver a juntarse (a reserva de que se produzca un fallo); las cadenas habilitadas no pueden juntarse con otras cadenas, ni terminar separadamente.

13.1.6 acción de ramificación permanente: Acción de división en la que las cadenas habilitadas no se volverán a juntar. Las cadenas habilitadas pueden interactuar y pueden terminar separadamente.

13.1.7 acción cabecera: En una actividad dada, una acción que no tiene predecesor.

13.1.8 subactividad: Subgráfico de una actividad que es a su vez una actividad y que cumple la condición de que, para cualquier par de acciones de ramificación temporal en la actividad progenitora, si se incluye en el subgráfico una de las dos acciones, hay que incluir también la otra.

13.2 Comportamiento contractual

Los conceptos descritos en esta cláusula se ilustran en la Figura 2.

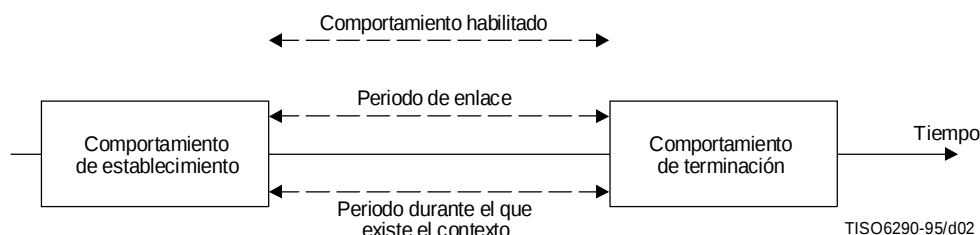


Figura 2 – Concepto de enlace y otros conceptos conexos

13.2.1 comportamiento de establecimiento: Comportamiento por el cual se establece un contrato dado entre determinados objetos. Un comportamiento de establecimiento puede ser:

- explícito, efectuado como resultado de interacciones de objetos que participarán en el contrato; o
- implícito, efectuado por una agencia externa (por ejemplo un objeto tercero, que no participa en el contrato), o que ha sido efectuado en una época anterior.

NOTAS

1 La negociación es un ejemplo de un género particular de comportamiento de establecimiento en el cual se intercambia información con miras a obtener una visión común de un comportamiento futuro permitido.

2 La publicación es un ejemplo de un género particular de comportamiento de establecimiento en el cual se distribuye información de un objeto a otros objetos.

3 Un comportamiento de establecimiento explícito tiene que incluir una instanciación de la plantilla asociada con el contrato. Esto puede seguir a una posible negociación/publicación que determinará qué contrato efectuar, qué plantilla instanciar, y con qué parámetros.

13.2.2 comportamiento habilitado: Comportamiento que caracteriza a un conjunto de objetos que se hace posible como resultado de un comportamiento de establecimiento.

El comportamiento habilitado no tiene que ser necesariamente el mismo para todos los objetos.

13.2.3 contexto contractual: El conocimiento de que un determinado contrato está en vigor y de que, por tanto, se requiere un determinado comportamiento de un conjunto de objetos.

Un objeto puede estar en varios contextos contractuales simultáneamente; en este caso, el comportamiento será la intersección de los comportamientos prescritos por cada contexto contractual.

NOTA – En OSI, el concepto de un contexto de presentación es un ejemplo de un contexto contractual que puede establecerse en el momento del establecimiento de la conexión, o posteriormente.

13.2.4 enlace: Relación entre un conjunto de objetos que se produce al efectuarse algún comportamiento de establecimiento; estado en el que se tiene en común un contexto contractual.

Un enlace se caracteriza por el comportamiento habilitado correspondiente.

NOTAS

1 Son ejemplos de enlaces que se producen como resultado de diferentes comportamientos de establecimiento:

- un diálogo [como en el procesamiento de transacciones en OSI (OSI-TP)];
- una vinculación (véase 13.4.2);
- una transacción distribuida [como en el procesamiento de transacciones en OSI (OSI-TP)];
- una conexión (N) (como en OSI);
- una asociación entre entidades (N) que las habilite para participar en comunicaciones sin conexión (N) (como en OSI);
- una relación entre unos ficheros y los procesos que ganan acceso a los ficheros.

2 Ciertos comportamientos pueden depender del establecimiento de múltiples enlaces conexos. Por ejemplo, una transacción distribuida puede depender tanto del enlace entre los usuarios de la transacción como de la asociación de soporte. El enlace entre los usuarios de la transacción (la transacción distribuida) puede continuar existiendo, pero estar inactiva, cuando se rompe la asociación.

3 Un enlace puede comprender más de dos objetos. No todos los objetos comprendidos en un enlace tienen necesariamente roles equivalentes. Por tanto, puede haber enlaces para la recogida o para la distribución de información. El número de participantes y los roles de los participantes los determina el contrato expresado por el enlace.

4 Hay una dualidad entre, por una parte, la aceptación de la especificación de un contexto contractual y de una obligación contractual y, por otra parte, el comportamiento habilitado. En la práctica, las estructuras pueden tener un nivel arbitrario de anidamiento, y un enlace de compromiso en un nivel puede también convenir un contrato que permita un enlace en un nivel más interior.

13.2.5 comportamiento de terminación: Comportamiento que rompe un enlace y repudia el contexto contractual y el contrato correspondientes.

Un comportamiento de terminación tiene que ser identificado explícitamente como tal en el contrato, si el comportamiento de establecimiento era explícito.

13.3 Causalidad

La identificación de causalidad permite la categorización de los roles de los objetos interactuantes. Esta cláusula presenta un conjunto básico de roles.

La causalidad implica una constricción sobre cada comportamiento de los objetos participantes mientras están interactuando. La causalidad se identificará en la definición de las clases (o de las subclases) a que pertenecen los objetos interactuantes, o en el refinamiento de las plantillas para sus clases (o subclases).

13.3.1 objeto iniciador (con respecto a una comunicación): Objeto que causa una comunicación.

NOTA – La identificación de un objeto iniciador con respecto a una comunicación implica una interpretación de la finalidad de la comunicación.

13.3.2 objeto respondedor: Objeto que participa en una comunicación y que no es el objeto iniciador.

13.3.3 objeto productor (con respecto a una comunicación): Objeto que es la fuente de la información transportada.

La utilización de este término no implica un determinado mecanismo de comunicación.

13.3.4 objeto consumidor (con respecto a una comunicación): Objeto que es un sumidero de la información transportada.

La utilización de este término no implica un determinado mecanismo de comunicación.

13.3.5 objeto cliente: Objeto que solicita que otro objeto realice una función.

13.3.6 objeto servidor: Objeto que realiza alguna función a nombre de un objeto cliente.

Pueden existir relaciones cliente/servidor de una naturaleza diferente (o de un nivel de abstracción diferente) entre un objeto y diferentes composiciones de los objetos con los que dicho objeto comunica.

NOTA – Informalmente, un servidor proporciona un servicio solicitado por un cliente.

13.4 Comportamientos de establecimiento

13.4.1 comportamiento de vinculación: Comportamiento de establecimiento entre dos o más interfaces (y por consiguiente entre sus objetos de soporte).

NOTA – «Vincular» significa «efectuar un comportamiento de vinculación».

13.4.2 vinculación: Contexto contractual, resultante de un comportamiento de establecimiento dado.

El comportamiento de establecimiento, el contexto contractual y el comportamiento habilitado pueden comprender sólo dos interfaces de objeto, o más de dos.

Un objeto que inicia un comportamiento de establecimiento puede participar o no en el subsiguiente comportamiento habilitado.

El comportamiento habilitado (y, por analogía, el contexto contractual) puede ser uniforme (cuando cada objeto participante puede hacer lo mismo que hacen todos los demás), o no uniforme (cuando un objeto participante tiene un rol diferente del de otro, como en el caso de los objetos cliente y servidor).

No hay una correspondencia necesaria entre un objeto que inicia un comportamiento de establecimiento y un rol particular en comportamientos habilitados no uniformes (por ejemplo, en un contexto contractual de cliente-servidor, cualquiera de los dos objetos puede haber iniciado validamente el comportamiento de establecimiento).

13.4.3 precondition de vinculación: Conjunto de condiciones que deben cumplirse para ejecutar con éxito un comportamiento de vinculación.

Los objetos que ejecutan el comportamiento de vinculación deben poseer los identificadores de todas las interfaces que intervienen en la vinculación. Puede haber precondiciones adicionales.

13.4.4 comportamiento de desvinculación: Comportamiento que termina una vinculación, o sea, el comportamiento de terminación para la vinculación.

13.4.5 comercio: Interacción entre objetos en la cual se intercambia información sobre contratos nuevos o potenciales a través de un objeto tercero. Comprende:

- a) *Exportación:* la provisión de un identificador a una interfaz, que deberá satisfacer algún enunciado de requisitos (por ejemplo, ofrecer un contrato potencial).
- b) *Importación:* la provisión de un identificador a una interfaz, que concuerda con un determinado enunciado de requisitos, lo que permite que se produzca un futuro comportamiento de vinculación (o sea, el establecimiento de un contrato).

13.5 Seguridad de funcionamiento

13.5.1 fallo: Infracción de un contrato.

NOTAS

1 El comportamiento especificado en el contrato es, por definición, el «comportamiento correcto». Un fallo es, por consiguiente, una desviación del cumplimiento del comportamiento correcto.

2 Las diferentes maneras en que un objeto puede fallar se denominan sus modos de fallo. Pueden distinguirse varios tipos de modos de fallo:

- fallos arbitrarios (incumplimiento de la especificación, que es el modo de fallo más general);
- fallo por omisión (cuando no se producen interacciones previstas);
- fallos fundamentales («crash failures») (fallos por omisión persistentes);
- fallos de temporización (funcionamiento incorrecto por no ajustarse a la base de tiempo).

3 Un mismo fallo puede ser percibido diferentemente por objetos diferentes en el entorno del objeto que lo sufre. Un fallo puede ser coherente, si todas las percepciones del fallo son iguales; incoherente, si los objetos en el entorno pueden tener percepciones diferentes de un mismo fallo.

13.5.2 error: Parte de un estado de objeto que pudiera provocar fallos. Una manifestación de una falta (véase 13.5.3) en un objeto.

NOTAS

1 El que un error conduzca o no un fallo depende de la descomposición del objeto, de su redundancia interna y del comportamiento del objeto. Una acción correctiva puede evitar que un error cause un fallo.

2 Un error puede estar latente (es decir, en un estado en que no es reconocido como tal), o detectado. Un error puede desaparecer antes de ser detectado.

13.5.3 falta (o avería, o defecto): Una situación que puede dar lugar a que se produzcan errores en un objeto.

NOTAS

1 Desde el momento en que se especifica un objeto hasta el momento en que es destruido pueden aparecer faltas que causen un error. Es posible que unas faltas que hayan aparecido en una época anterior (por ejemplo, faltas cometidas en el diseño) no provoquen un fallo sino en una época posterior (por ejemplo, en la ejecución).

2 Una falta puede estar activa o «durmiendo». Una está activa cuando produce errores. La presencia de faltas activas sólo se determina por la detección de errores.

3 Las faltas pueden ser:

- accidentales (cuando aparecen o se crean de modo fortuito) o intencionales (cuando se crean voluntariamente);
- físicas (cuando se deben a algún fenómeno físico) o antropogénicas (cuando provienen de una actividad humana);
- internas (cuando son parte de un estado de objeto que puede causar un error) o externas (cuando provienen de la interferencia o interacción con el entorno);
- permanentes o temporales.

4 Las definiciones de falta, error y fallo implican, recursivamente, dependencias causales entre faltas, errores y fallos:

- una falta puede provocar un error (provocará un error si se torna activa);
- un error puede provocar un fallo de sistema (provocará un fallo de sistema a menos que el sistema pueda tratarlo);
- se produce un fallo cuando un error afecta al funcionamiento correcto del servicio suministrado por un sistema (o por un componente de un sistema).

13.5.4 estabilidad: Propiedad que tiene un objeto, con respecto a un determinado modo de fallo, cuando no es afectado por ese modo de fallo.

14 Conceptos de gestión

La gestión en procesamiento abierto distribuido (ODP) se refiere a la gestión de sistemas globales, incluidas la gestión de aplicaciones y la gestión de comunicaciones.

14.1 gestión de aplicaciones: Gestión de aplicaciones dentro de un sistema ODP. Algunos aspectos de la gestión de aplicaciones son comunes a todas las aplicaciones y se conocen por la gestión independiente de la aplicación. Los aspectos que son específicos de una aplicación dada se conocen por gestión específica de aplicación.

14.2 gestión de comunicaciones: Gestión de objetos que soportan la comunicación entre objetos dentro de un sistema de procesamiento abierto distribuido.

14.3 información de gestión: Conocimiento relativo a objetos que son de interés para la gestión.

14.4 rol gestionado: La visión de la interfaz de gestión de un objeto que está siendo gestionado dentro de un sistema de procesamiento abierto distribuido.

NOTA – Cuando el objeto proporciona servicios de comunicación OSI, la gestión OSI se refiere a la interfaz gestionada como un objeto gestionado.

14.5 rol gestor: La visión de un objeto que está realizando acciones de gestión.

14.6 notificación: Interacción iniciada por un objeto que funciona en un rol gestionado.

15 Enfoque de la conformidad atendiendo al ODP

15.1 Conformidad con las normas ODP

La conformidad relaciona una implementación con una norma. Toda proposición que sea verdadera en la especificación tiene que ser verdadera en la implementación.

Un enunciado de conformidad es un enunciado que identifica puntos de conformidad de una especificación y el comportamiento que debe cumplirse en esos puntos. Los enunciados de conformidad aparecerán solamente en normas destinadas a constreñir alguna característica de una implementación real, por lo que, en principio, existe la posibilidad de prueba.

El modelo de referencia de procesamiento distribuido abierto (RM-ODP) identifica ciertos puntos de referencia en la arquitectura como potencialmente declarables como puntos de conformidad en especificaciones, es decir, como puntos en los cuales se puede probar la conformidad, y, por consiguiente, como puntos que tendrán que ser accesibles para la prueba. Sin embargo, el requisito de que un determinado punto de referencia se considere un punto de conformidad deberá enunciarse explícitamente en el enunciado de conformidad de la especificación en cuestión.

Los requisitos que aseguran que un miembro de la familia de normas ODP tenga necesariamente que ser coherente con otro (como el RM-ODP) se establecen durante el proceso de normalización. La satisfacción de estos requisitos se llama **cumplimiento**.

Si una especificación cumple directa o indirectamente algunas otras normas, las proposiciones que son verdaderas en esas normas son también verdaderas en una implementación conforme de la especificación.

15.2 Pruebas y puntos de referencia

La veracidad de un enunciado en una implementación sólo puede determinarse por medio de pruebas y se basa en una correspondencia de términos en la especificación con aspectos observables de la implementación.

En cualquier nivel dado de abstracción, una prueba es una serie de estímulos y eventos observables, efectuada en puntos prescritos, llamados puntos de referencia, y solamente en estos puntos. Estos puntos de referencia son interfaces accesibles. Un componente de sistema con respecto al cual se pretende la conformidad se percibe como una «caja negra», que puede probarse solamente en sus conexiones externas. Así, por ejemplo, la conformidad con especificaciones de protocolo OSI no depende de ninguna estructura interna del sistema sometido a prueba.

15.3 Clases de puntos de referencia

Un punto de conformidad es un punto de referencia en el que puede efectuarse una prueba para saber si un sistema satisface un conjunto de criterios de conformidad. Un enunciado de conformidad tiene que especificar dónde están los puntos de conformidad, y qué criterios se satisfacen en esos puntos. Se definen cuatro clases de puntos de referencia en los que pueden efectuarse pruebas de conformidad.

15.3.1 punto de referencia programática: Punto de referencia en el que puede establecerse una interfaz programática para permitir el acceso a una función. Un requisito de conformidad programática se enuncia en términos de una compatibilidad en comportamiento con la finalidad de que un objeto sea reemplazado por otro. Una interfaz programática es una interfaz que está realizada mediante la vinculación de un lenguaje de programación.

NOTA – Por ejemplo, se puede establecer un punto de referencia programática en una norma de base de datos para soportar una vinculación de lenguaje en algún nivel de abstracción.

15.3.2 punto de referencia perceptual: Punto de referencia en el que hay alguna interacción entre el sistema y el mundo físico.

NOTAS

1 Un punto de referencia perceptual puede ser, por ejemplo, una interfaz hombre-computador o una interfaz robótica (especificada en términos de las interacciones del robot con su entorno físico).

2 Un requisito de conformidad perceptual de la interfaz hombre-computador se enuncia en términos de la forma de la información presentada a las personas y de la metáfora de interacción y los diálogos en que pueden intervenir las personas.

3 Un punto de referencia perceptual puede establecerse, por ejemplo, en una norma gráfica.

15.3.3 punto de referencia de interfuncionamiento: Punto de referencia en el que puede establecerse una interfaz para permitir la comunicación entre dos o más sistemas. Un requisito de conformidad de interfuncionamiento se enuncia en términos del intercambio de información entre dos o más sistemas. La conformidad de interfuncionamiento comprende la interconexión de puntos de referencia.

NOTA – Por ejemplo, las normas OSI se basan en la interconexión de puntos de referencia de interfuncionamiento (el medio físico).

15.3.4 punto de referencia de intercambio: Punto de referencia en el que puede introducirse, en el sistema, un medio físico externo de almacenamiento. Un requisito de conformidad de intercambio se enuncia en términos del comportamiento (métodos de acceso y formatos) de algún medio físico de tal modo que la información pueda registrarse en un sistema, y transferirse físicamente, directa o indirectamente, para utilizarla en otro sistema.

NOTA – Por ejemplo, algunas normas de intercambio de información se basan en puntos de referencia de intercambio.

15.4 Cambio de configuración

La prueba de conformidad puede efectuarse en un solo punto de referencia o puede implicar cierto grado de coherencia al utilizarse en una serie de configuraciones que comprenden varios puntos de referencia. Esto puede incluir la aplicación de pruebas de conformidad:

- a) al requisito de que un componente pueda funcionar después de algún proceso preparatorio que lo adapte al entorno local;
- b) al requisito de que un componente funcione de acuerdo con su especificación en un determinado punto de referencia en todo momento, desde su inicialización;
- c) al requisito de que un componente continúe funcionando cuando sea trasladado a un entorno similar en el curso de una operación.

Las propiedades probadas antes mencionadas dan lugar a los siguientes atributos de los objetos o interfaces que intervienen.

15.4.1 portabilidad: Propiedad de que los puntos de referencia de un objeto permitan adaptarlo a una diversidad de configuraciones.

NOTA – Si el punto de referencia es un punto de referencia programática, el resultado puede ser la portabilidad del código fuente o de la ejecución. Si es un punto de referencia de interfuncionamiento, el resultado es la portabilidad del equipo.

15.4.2 trasladabilidad: Aptitud para cambiar la configuración, sustituyendo un punto de referencia de un objeto por otro mientras se está utilizando el objeto.

15.5 Proceso de prueba de conformidad

La conformidad es un concepto que puede aplicarse en cualquier nivel de abstracción. Por ejemplo, cabe esperar que a una norma que defina los tipos de caracteres se aplique una conformidad perceptual muy detallada, y que, en cambio, a las reglas sobre la organización de la pantalla se aplique una conformidad perceptual mucho más abstracta.

Cuanto más abstracta sea una especificación, tanto más difícil es su prueba. Se necesita una mayor cantidad de interpretación específica de la implementación para establecer que las proposiciones más abstractas sobre las implementaciones son efectivamente verdaderas. No es evidente que la prueba directa de especificaciones muy abstractas sea posible a un costo razonable utilizando técnicas actualmente disponibles o previsibles.

El proceso de prueba hace referencia a una especificación. Para estar completa, una especificación debe contener:

- a) el comportamiento del objeto que se está normalizando y la manera de obtener dicho comportamiento;
- b) una lista de los términos primitivos utilizados en la especificación cuando se formulan los enunciados de comportamiento;
- c) un enunciado de conformidad que indica los puntos de conformidad, lo que las implementaciones tienen que hacer en ellos y la información que los implementadores deben suministrar (esto corresponde a las nociones OSI de PICS y PIXIT).

En el proceso de prueba hay dos roles: implementador y probador. El primer rol es el del implementador, que efectúa una implementación sobre la base de una especificación. El implementador proporcionará un enunciado de la correspondencia de todos los términos utilizados en la especificación con cosas o sucesos en el mundo real. Por tanto, hay que indicar las interfaces correspondientes a los puntos de conformidad y dar la representación de las señales. Si la especificación es abstracta, la correspondencia de sus términos básicos con el mundo real puede ser en sí compleja. Por ejemplo, en una especificación de punto de vista computacional (véase la Rec. UIT-T X.903 | ISO/CEI 10746-3), los términos primitivos podrían ser un conjunto de interacciones entre objetos. El implementador que deseara ajustarse a la especificación de punto de vista computacional tendría que indicar cómo se proporcionaron las interacciones, sea por referencia a una especificación de ingeniería o proporcionando una descripción detallada de un mecanismo no normalizado (aunque este procedimiento limite el campo de aplicación de la implementación a sistemas en que existe el acuerdo de utilizar el mecanismo no normalizado).

El segundo rol es el del probador, que observa el sistema sometido a prueba. La prueba implica cierto comportamiento compartido entre el proceso de prueba y el sistema sometido a prueba. Si este comportamiento se da con un etiquetado causal, hay todo un abanico de tipos de pruebas que va de:

- a) la prueba pasiva, en la que todo el comportamiento es originado por el sistema sometido a prueba y registrado por el probador;
- b) a la prueba activa, en la que el comportamiento es originado y registrado por el probador.

Normalmente, la especificación del sistema sometido a prueba es en forma de una interfaz, como en la especificación del probador y de los procedimientos de prueba. Cuando se efectúa la prueba, los interfaces están vinculados.

El probador interpretará sus observaciones utilizando la correspondencia proporcionada por el implementador para obtener proposiciones sobre la implementación, las cuales podrán entonces comprobarse para determinar que son también verdaderas en la especificación.

15.6 Resultado de las pruebas

El proceso de prueba tiene éxito si todas las comprobaciones con respecto a la especificación tienen éxito. En cambio, puede fracasar si se da cualquiera de los casos siguientes:

- a) la especificación es lógicamente incoherente o incompleta, por lo que las proposiciones sobre la implementación no pueden comprobarse (este caso nunca debería ocurrir);
- b) la correspondencia proporcionada por el implementador es lógicamente incompleta, por lo que es incoherente, o las observaciones no pueden relacionarse con términos de la especificación; la prueba es imposible;
- c) el comportamiento observado no puede interpretarse de acuerdo con la correspondencia dada por el implementador. El comportamiento del sistema no tiene sentido en términos de la especificación, y la prueba fracasa por esta razón;
- d) el comportamiento se interpreta para que dé términos expresados en la especificación, pero la forma en que aparecen estos términos da proposiciones que no son verdaderas en la especificación, y la prueba fracasa por esta razón.

15.7 Relación entre puntos de referencia

El flujo de información entre los componentes del sistema modelado pueden pasar a través de más de un punto de referencia. Por ejemplo, un sistema distribuido puede comprender interacciones de dos componentes A y B, pero la comunicación entre ellos puede, a su vez, pasar a través de una interfaz programática, un punto de interconexión y una ulterior interfaz programática.

Un refinamiento del mismo sistema puede mostrar componentes interconectados que tienen más de un componente en el trayecto de comunicación entre ellos.

En ambos casos, la prueba de conformidad puede comprender:

- a) la prueba del flujo de información en cada uno de estos puntos de referencia;
- b) la prueba de la coherencia entre los eventos en pares de puntos de referencia.

En general, la prueba del comportamiento correcto de una configuración de objetos requerirá la prueba de que los enunciados sobre las interfaces de comunicación son verdaderos, pero requerirá también la observación de otras interfaces de estos objetos, de modo que se puedan comprobar también los enunciados sobre la composición.

La noción general de conformidad tiene en cuenta la relación entre varios puntos de conformidad. Dado que la especificación relacionada con un punto de conformidad dado puede expresarse en varios niveles de abstracción, la prueba en un punto de conformidad dado comprenderá siempre una interpretación en el nivel de abstracción apropiado. Así, la prueba del comportamiento global requiere una prueba coordinada en todos los puntos de conformidad que intervienen, y la utilización de la interpretación apropiada en cada punto.

En particular, sólo puede afirmarse la certeza de la conformidad de una plantilla con una determinada interfaz programática cuando se considera la vinculación (de lenguaje) para el lenguaje en que se ha escrito la plantilla, y que la planilla escrita se ajusta a la especificación de la vinculación de lenguaje, la que, a su vez, tiene que ser conforme con la especificación de interfaz abstracta.

Índice alfabético

NOTA – Para cada uno de los artículos de este índice alfabético se indica el número de la cláusula o subcláusula en que está definido.

- abstracción, 6.3
- acción, 8.3
- acción cabecera, 13.1.7
- acción de denominación, 12.5
- acción de división, 13.1.4
- acción de junción, 13.1.3
- acción de ramificación permanente, 13.1.6
- acción de ramificación temporal, 13.1.5
- acción interna, 8.3
- actividad, 8.5
- arquitectura (de un sistema), 6.6
- atomicidad, 6.4
- avería (véase falta)
- cadena (de acciones), 13.1.1
- calidad de servicio, 11.2.2
- clase (de <Xs>), 9.8
- clase de base, 9.21
- clase (de) plantilla (de una <X>), 9.20
- clase derivada, 9.21
- comercio, 13.4.5
- compatibilidad en comportamiento, 9.4
- compatibilidad en comportamiento coercida, 9.4
- compatibilidad en comportamiento natural, 9.4
- comportamiento (de un objeto), 8.6
- comportamiento de establecimiento, 13.2.1
- comportamiento de terminación, 13.2.5
- comportamiento de vinculación, 13.4.1
- comportamiento habilitado, 13.2.2
- composición, 9.1
- comunicación, 8.8
- configuración (de objetos), 10.2
- contexto contractual, 13.2.3
- contexto de denominación, 12.4
- contrato, 11.2.1
- contrato de entorno, 11.2.3
- creación (de una <X>), 9.15
- cumplimiento, 15.1
- datos, 3.2.6
- defecto (véase falta)
- descomposición, 9.3
- desvinculación, 13.4.4
- dominio <X>, 10.3
- dominio de denominación, 12.6
- enlace («liaison»), 13.2.4
- entidad, 6.1
- entorno (de un objeto), 8.2
- época, 10.5
- error, 13.5.2
- espacio de nombre, 12.3
- estabilidad, 13.5.4
- estado (de un objeto), 8.7
- fallo, 13.5.1
- falta, 13.5.3
- firma de interfaz, 9.12
- gestión de aplicaciones, 14.1
- gestión de comunicaciones, 14.2
- gráfico de denominación, 12.7
- grupo <X>, 10.1
- hilo, 13.1.2
- identificador, 12.2
- información, 3.2.5
- información de gestión, 14.3
- instancia (de un tipo), 9.18
- instanciación (de una plantilla <X>), 9.13
- interacción, 8.3
- interfaz, 8.4
- introducción (de un <X>), 9.16
- invariante, 9.22
- isocronicidad, 11.3.2
- nombre, 12.1
- normas ODP, 3.2.2
- notificación, 14.6
- objeto, 8.1

objeto cliente, 13.3.5
 objeto compuesto, 9.2
 objeto consumidor, 13.3.4
 objeto iniciador, 13.3.1
 objeto productor, 13.3.3
 objeto respondedor, 13.3.2
 objeto servidor, 13.3.6
 obligación, 11.2.4
 oración, 7.2
 papel (véase rol)
 permiso, 11.2.5
 persistencia, 11.3.1
 plantilla <X>, 9.11
 política, 11.2.7
 portabilidad, 15.4.1
 postcondición, 9.24
 precondition, 9.23
 precondition de vinculación, 13.4.3
 procesamiento distribuido, 3.2.1
 procesamiento distribuido abierto, 3.2.3
 prohibición, 11.2.6
 proposición, 6.2
 punto de conformidad, 10.7
 punto de interacción, 8.11
 punto de referencia, 10.6
 punto de referencia de intercambio, 15.3.4

punto de referencia de interfuncionamiento, 15.3.3
 punto de referencia perceptual, 15.3.2
 punto de referencia programática, 15.3.1
 punto de vista (de un sistema), 3.2.7
 rastro, 9.6
 refinamiento, 9.5
 resolución de nombre, 12.8
 rol, 9.14
 rol gestionado, 14.4
 rol gestor, 14.5
 sistema, 6.5
 sistema ODP, 3.2.4
 subactividad, 13.1.8
 subclase, 9.10
 subdominio, 10.4
 subtipo, 9.9
 superclase, 9.10
 supertipo, 9.9
 supresión (de una <X>), 9.17
 término, 7.1
 tipo (de una <X>), 9.7
 tipo de plantilla (de una <X>), 9.19
 transparencia de distribución, 11.1.1
 trasladabilidad, 15.4.2
 ubicación en el espacio, 8.9
 ubicación en el tiempo, 8.10
 vinculación, 13.4.2