



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.952

(12/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Procesamiento distribuido abierto

**Tecnología de la información – Procesamiento
distribuido abierto – Función intermediación:
Provisión de la función intermediación a través
del servicio de directorio de interconexión de
sistemas abiertos**

Recomendación UIT-T X.952

(Anteriormente Recomendación del CCITT)

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

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 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
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–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.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
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
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
SEGURIDAD	
X.800–X.849	
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

Para más información, véase la Lista de Recomendaciones del UIT-T.

NORMA INTERNACIONAL 13235-3

RECOMENDACIÓN UIT-T X.952

TECNOLOGÍA DE LA INFORMACIÓN – PROCESAMIENTO DISTRIBUIDO ABIERTO – FUNCIÓN DE INTERMEDIACIÓN: PROVISIÓN DE LA FUNCIÓN INTERMEDIACIÓN A TRAVÉS DEL SERVICIO DE DIRECTORIO DE INTERCONEXIÓN DE SISTEMAS ABIERTOS

Resumen

La presente Recomendación | Norma Internacional describe cómo se puede realizar la función intermediación de procesamiento distribuido abierto (ODP) utilizando asientos de información y mecanismos de soporte del directorio de interconexión de sistemas abiertos (OSI). Esta Especificación se ha de utilizar junto con la norma sobre la función intermediación de ODP (véase la Rec. UIT-T X.950 | ISO/CEI 13235-1). En caso de discrepancias entre los enunciados prescriptivos de la Rec. UIT-T X.950 y los de la presente Especificación, los enunciados de la Rec. X.950 tienen precedencia.

La presente Especificación abarca:

- las plantillas normalizadas para objetos de información de la función intermediación en el árbol de información del directorio;
- las descripciones de correspondencia entre las operaciones de la función intermediación y las operaciones apropiadas del directorio;
- la descripción de la utilización de otras características del directorio para proporcionar mecanismos de soporte con miras a implementar la función intermediación de ODP.

El campo de aplicación de la presente Especificación es la construcción de la función intermediación de ODP utilizando el directorio de OSI, cuando proceda.

Orígenes

El texto de la Recomendación UIT-T X.952 se aprobó el 12 de diciembre de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 13235-3.

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 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

Página

1	Alcance y campo de aplicación	1
2	Referencias normativas	1
	2.1 Recomendaciones Normas Internacionales idénticas	1
3	Definiciones	2
4	Abreviaturas	4
5	Visión general	4
6	Esquema	5
	6.1 Generalidades	6
	6.2 Asiento de intermediario (trader Entry)	7
	6.2.1 Nombre común (commonName).....	7
	6.2.2 Interfaz de intermediario (traderInterface).....	8
	6.2.3 Nombre de DSA (dsaName)	8
	6.2.4 Depositario de tipos (typeRepos).....	8
	6.2.5 Tarjeta de búsqueda por defecto (defSearchCard)	8
	6.2.6 Tarjeta de búsqueda máxima (maxSearchCard).....	8
	6.2.7 Tarjeta de concordancia por defecto (defMatchCard)	9
	6.2.8 Tarjeta de concordancia máxima (maxMatchCard)	9
	6.2.9 Tarjeta de devolución por defecto (defReturnCard)	9
	6.2.10 Tarjeta de devolución máxima (maxReturnCard)	9
	6.2.11 Cómputo de saltos por defecto (defHopCount).....	10
	6.2.12 Cómputo de saltos máximo (maxHopCount).....	10
	6.2.13 Política de seguimiento por defecto (defFollowPolicy)	10
	6.2.14 Política de seguimiento máximo (maxFollowPolicy).....	11
	6.2.15 Política de seguimiento de enlace máximo (maxLinkFollowPolicy)	11
	6.2.16 Admite propiedades modificables (supportsModifiableProperties)	11
	6.2.17 Admite propiedades dinámicas (supportsDynamicProperties).....	11
	6.2.18 Admite ofertas por procuración (supportsProxyOffers).....	12
	6.2.19 Lista máxima (maxList).....	12
	6.2.20 Petición de origen de identificador (requestIdStem)	12
	6.2.21 Descripción (description).....	12
	6.2.22 Contraseña de usuario (userPassword).....	12
	6.2.23 Otros atributos X.500.....	12
	6.3 Asiento de políticas de intermediario (traderPolicyEntry)	13
	6.3.1 Nombre común (commonName).....	13
	6.3.2 Constricción de gestión de tipos (typeManagementConstraint)	13
	6.3.3 Constricción de búsqueda (searchConstraint)	14
	6.3.4 Constricción de aceptación de oferta (offerAcceptanceConstraint)	14
	6.3.5 Otros atributos X.500.....	14
	6.4 Asiento de oferta de servicios (serviceOfferEntry)	14
	6.4.1 Identificador de oferta de servicio (sOfferId)	15
	6.4.2 Identificador de interfaz de servicio (serviceInterfaceId)	16
	6.4.3 Identificador de tipo de servicio (serviceTypeId)	16
	6.4.4 Tiene propiedades dinámicas (hasDynamicProperties).....	16
	6.4.5 Tiene propiedades modificables (hasModifiableProperties)	17
	6.4.6 Propiedades dinámicas (dynamicProps)	17
	6.4.7 Otros atributos X.500.....	17
	6.5 Asiento de enlace de intermediarios (traderLinkEntry)	18
	6.5.1 Nombre de enlace (linkName)	18
	6.5.2 Identificador de enlace (linkId).....	18
	6.5.3 Identificador de interfaz de intermediario deseado (targetTraderInterfaceId)	19
	6.5.4 Regla de seguimiento de transferencia por defecto (defPassOnFollowRule).....	19
	6.5.5 Regla de limitadora de seguimiento (limitingFollowRule)	19
	6.5.6 Otros atributos X.500.....	19

	<i>Página</i>	
6.6	Asiento de oferta por procuración (proxyOfferEntry)	20
6.6.1	Identificador de oferta por procuración (proxyOfferId)	20
6.6.2	Identificador de interfaz de consulta por procuración (proxyLookUpInterfaceId)	21
6.6.3	Fórmula de restricción (constraintRecipe)	21
6.6.4	Si todo concuerda (ifMatchAll)	21
6.6.5	Otros atributos X.500.....	21
6.7	Otros asientos X.500 utilizados por el T-DUA	22
7	Operaciones.....	22
7.1	Inicialización.....	23
7.2	Operaciones de clientes	23
7.3	Operaciones de registro	23
7.3.1	Exportar	24
7.3.2	Retirar	25
7.3.3	Modificar	25
7.3.4	Describir	26
7.3.5	Retirar con restricción	27
7.3.6	Resolver	27
7.4	Operaciones consultar.....	28
7.4.1	Operación indagar.....	28
7.4.2	Políticas	28
7.4.3	Búsqueda local.....	29
7.4.4	Búsqueda de intermediario federados	30
7.4.5	Búsqueda de ofertas por procuración	30
7.4.6	Oferta de servicio devuelta	30
7.5	Operaciones de enlace	31
7.5.1	Añadir enlace	31
7.5.2	Suprimir enlace	31
7.5.3	Modificar enlace	32
7.5.4	Describir enlace	32
7.5.5	Enumerar enlaces.....	33
7.6	Operaciones de ofertas por procuración	33
7.6.1	Exportar por procuración	33
7.6.2	Retirar por procuración	34
7.6.3	Describir por procuración	34
7.7	Operaciones de atributos de intermediarios	35
7.8	Operaciones administrativas	35
7.8.1	Enumerar ofertas.....	35
7.8.2	Enumerar procuraciones	36
7.9	Operaciones de evaluación de propiedades dinámicas	36
7.9.1	Evaluar propiedades dinámicas.....	37
8	Depositario de tipos	37
8.1	Esquema X.500 y el depositario de tipos mínimo.....	37
9	Propiedades dinámicas	38
9.1	Exportación de una oferta de servicio.....	38
9.2	Importación de una oferta de servicio.....	39
	Anexo A – Definición esquemática de definiciones de intermediario.....	40
	Anexo B – Ejemplo de definición esquemática de descripción de servicio	50

Introducción

La función intermediación del procesamiento distribuido abierto (ODP, *open distributed processing*) (véanse las Recomendaciones de la serie UIT-T X.950 | ISO/CEI 13235) proporciona el medio de ofrecer un servicio y de descubrir servicios que han sido ofrecidos. La Rec. UIT-T X.950 | ISO/CEI 13235-1 define una especificación de empresa, una especificación de información y una especificación computacional de esta función intermediación, pero no define ninguna especificación de ingeniería. La presente Recomendación | Norma Internacional describe cómo se pueden diseñar las especificaciones de la función intermediación en la Rec. UIT-T X.950 | ISO/CEI 13235-1 utilizando el servicio de directorio de interconexión de servicios abiertos (OSI, *open systems interconnection*) (véase la Rec. UIT-T X.500 | ISO/CEI 9594-1) para almacenar información y proporcionar mecanismos de soporte. La presente especificación no prescribe que un intermediario debe ser realizado utilizando el directorio de OSI. Sin embargo, si se utiliza el directorio de OSI, la presente Especificación define las plantillas normalizadas para los asientos de información (por ejemplo, los objetos oferta de servicios e información de enlaces) en el árbol de información del directorio.

La cláusula 5 da una visión general de cómo se implementa la función intermediación como una combinación del agente de usuario del directorio y del agente de sistema del directorio de la Recomendación X.500. El agente de sistema del directorio de la Recomendación X.500 se utiliza para almacenar el objeto información de intermediario y un agente de usuario del directorio del intermediario implementa la funcionalidad requerida por un intermediario, que es difícil o imposible llevar a cabo utilizando los servicios del directorio de OSI.

La cláusula 6 define las plantillas normalizadas para los asientos de información del objeto información de intermediario, la información conocida por un intermediario determinado.

La cláusula 7 describe la correspondencia entre las operaciones de la función intermediación y las operaciones apropiadas del directorio.

La cláusula 8 especifica una función depositaria de tipos mínima necesaria para permitir el funcionamiento correcto del directorio de la Recomendación X.500 para intermediación.

La cláusula 9 describe los mecanismos utilizados para poder efectuar el tratamiento de las propiedades dinámicas de las ofertas de servicio del intermediario.

La presente Especificación tiene dos anexos.

El anexo A es una definición esquemática normativa de las definiciones de intermediario.

El anexo B es una definición esquemática informativa de un ejemplo de descripción de servicio.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

**TECNOLOGÍA DE LA INFORMACIÓN – PROCESAMIENTO DISTRIBUIDO
ABIERTO – FUNCIÓN DE INTERMEDIACIÓN: PROVISIÓN DE LA FUNCIÓN
INTERMEDIACIÓN A TRAVÉS DEL SERVICIO DE DIRECTORIO
DE INTERCONEXIÓN DE SISTEMAS ABIERTOS**

1 Alcance y campo de aplicación

La presente Especificación describe cómo se puede realizar la función intermediación de ODP utilizando asientos de información y mecanismos de soporte del directorio de OSI. Esta Especificación se ha de utilizar junto con la norma sobre la función intermediación de ODP (Rec. UIT-T X.950 | ISO/CEI 13235-1). En caso de discrepancia entre los enunciados prescriptivos de la Rec. UIT-T X.950 y los de la presente Especificación, los enunciados de la Rec. UIT-T X.950 ISO/CEI 13235-1 tienen precedencia.

La presente Especificación abarca:

- las plantillas normalizadas para objetos de información de la función intermediación en el árbol de información del directorio;
- las descripciones de la correspondencia de operaciones entre la función intermediación y operaciones apropiadas del directorio;
- descripción de la utilización de otras características del directorio para proporcionar mecanismos de soporte con miras a implementar la función intermediación de ODP.

La presente Especificación no prescribe que un intermediario debe ser diseñado utilizando el directorio de OSI. Sin embargo, si se utiliza el directorio de OSI, esta Especificación define plantillas normalizadas para los asientos de información (por ejemplo, los objetos oferta de servicio e información de enlace) en el árbol de información del directorio. Esta Especificación no impone restricciones sobre dónde se colocan estos asientos en el árbol de información del directorio. Es decir, esta Especificación no normaliza reglas de estructura, pero sí describe un mecanismo para proporcionar la función intermediación utilizando el directorio de OSI.

El campo de aplicación de la presente Especificación es la construcción de la función intermediación de ODP utilizando el directorio de OSI, cuando proceda.

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.500 (1993) | ISO/CEI 9594-1:1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio – Visión de conjunto de conceptos, modelos y servicios*.
- Recomendación UIT-T X.501 (1993) | ISO/CEI 9594-2:1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos*.
- Recomendación UIT-T X.509 (1993) | ISO/CEI 9594-8:1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio – Marco de autenticación*.

- Recomendación UIT-T X.511 (1993) | ISO/CEI 9594-3:1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio – Definición del servicio abstracto*.
- Recomendación UIT-T X.519 (1993) | ISO/CEI 9594-5: 1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolo*.
- Recomendación UIT-T X.520 (1993) | ISO/CEI 9594-6:1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados*.
- Recomendación UIT-T X.521 (1993) | ISO/CEI 9594-7: 1995, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Clases de objeto seleccionadas*.
- Recomendación UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica*.
- Recomendación UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de objetos de información*.
- Recomendación UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones*.
- Recomendación UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de las especificaciones de la notación de sintaxis abstracta uno*.
- 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.950 (1997) | ISO/CEI 13235-1¹⁾, *Tecnología de la información – Procesamiento distribuido abierto – Función intermediación: Especificaciones*.

3 Definiciones

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.902 | ISO/CEI 10746-2:

- actividad;
- comportamiento;
- objeto cliente;
- fallo;
- identificador;
- instancia;
- interacción;
- interfaz;
- signatura de interfaz;
- nombre;
- objeto;
- obligación;
- sistema ODP;
- política;
- objeto servidor;
- subtipo;
- plantilla <X>;
- intermediación;

¹⁾ Por publicar.

- tipo;
- punto de vista.

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.903 | ISO/CEI 10746-3:

- administrador;
- comunidad;
- punto de vista computacional;
- referencia de interfaz de ingeniería;
- punto de vista de la ingeniería;
- punto de vista de la empresa;
- exportador;
- importador;
- punto de vista de la información;
- exportación de servicio;
- importación de servicio;
- oferta de servicio;
- punto de vista de la tecnología;
- función intermediación;
- función depositaria de tipos.

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.950 | ISO/CEI 13235-1:

- intermediarios federados;
- iterador;
- enlace;
- oferta por procuración;
- tipo de servicio;
- propiedad del servicio;
- intermediario;
- atributo de intermediario;
- gráfico de intermediación.

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.500 | ISO/CEI 9594-1:

- directorio;
- base de información del directorio;
- usuario (del directorio).

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- atributo;
- tipo de atributo;
- valor de atributo;
- árbol de información del directorio;
- agente de sistema del directorio;
- agente de usuario de directorio;
- nombre distinguido;
- asiento (o inserción) (del directorio);
- filtro;

ISO/CEI 13235-3 : 1998 (S)

- regla de concordancia;
- nombre (del directorio);
- forma de nombre;
- objeto;
- clase de objeto;
- asiento de objeto;
- nombre distinguido relativo;
- regla de estructura;
- subclase;
- subordinado;
- superclase.

En esta Recomendación | Norma Internacional se utilizan las siguientes operaciones definidas en la Rec. UIT-T X.511 | ISO/CEI 9594-3:

- añadir asiento;
- modificar asiento;
- leer;
- suprimir asiento;
- buscar.

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.509 | ISO/CEI 9594-8:

- autenticación;
- contraseña.

4 Abreviaturas

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

DIB	Base de información del directorio (<i>directory information base</i>)
DIT	Árbol de información del directorio (<i>directory information tree</i>)
DN	Nombre distinguido (<i>distinguished name</i>)
DSA	Agente de sistema del directorio (<i>directory system agent</i>)
DUA	Agente de usuario del directorio (<i>directory user agent</i>)
ODP	Procesamiento distribuido abierto (<i>open distributed processing</i>)
OID	Identificador de objeto (<i>object identifier</i>)
RDN	Nombre distinguido relativo (<i>relative distinguished name</i>)
T-DUA	Agente de usuario del directorio de intermediario (<i>trader directory user agent</i>)

5 Visión general

En la presente Especificación, la función intermediación se implementa como una combinación de DUA y DSA de la Recomendación X.500. En la medida posible, las características de la Recomendación X.500 se utilizan para implementar directamente la función intermediación, pero no todas las características de intermediario pueden ser sustentadas directamente por la Recomendación X.500. Por este motivo, el intermediario (el objeto que proporciona la función intermediación) está formado por dos componentes: un directorio X.500 que almacena la información del intermediario y un DUA de intermediario (T-DUA) que implementa la funcionalidad requerida por un intermediario que es difícil, o imposible de realizar en el marco de la Recomendación X.500. El directorio X.500 se utiliza para almacenar el objeto información de intermediario. Las peticiones de los clientes del intermediario (importadores y exportadores) corresponden a operaciones en la base de datos de la Recomendación X.500. La figura 1 muestra los componentes de un intermediario y sus interacciones con sus clientes.

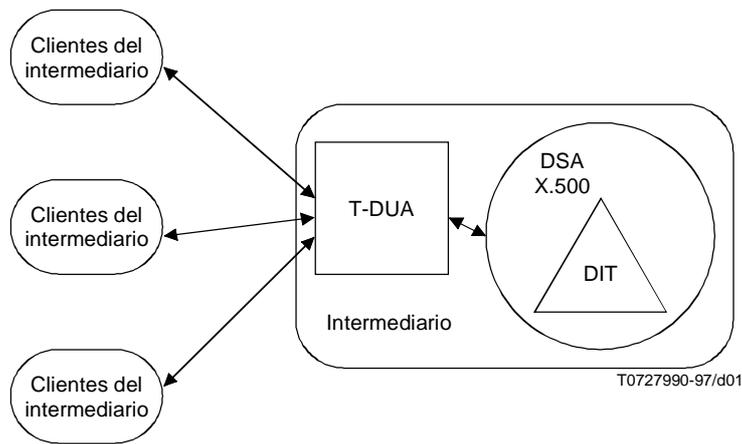


Figura 1 – El intermediario con sus componentes y clientes

El T-DUA y los clientes del intermediario (importadores y exportadores) comunican utilizando un protocolo de intermediario, que no se define en la presente especificación. Puede ser cualquier protocolo que aplique la funcionalidad especificada por la Rec. UIT-T X.950 | ISO/CEI 13235-1. La finalidad de esta Especificación es especificar cómo el T-DUA utiliza un directorio X.500 para apoyar la funcionalidad especificada por la Rec. UIT-T X.950 | ISO/CEI 13235-1.

La información almacenada por el directorio X.500 comprende:

- Los atributos del intermediario (es decir, información sobre el propio intermediario).
- Las políticas de empresa del intermediario (es decir, las reglas para determinar y orientar el comportamiento del intermediario).
- El conjunto de ofertas de servicio (es decir, la información utilizada por el intermediario cuando actúa como un servidor).
- El conjunto de enlaces de intermediario (es decir, la información utilizada por el intermediario cuando actúa como un cliente).
- El conjunto de ofertas por procuración (es decir, la información utilizada por el intermediario cuando actúa como un servidor para ofertas por procuración).

La Recomendación X.500 se utiliza para almacenar esta información por varios motivos:

- El modelo de información requerido por el intermediario de ODP es muy similar al proporcionado por la Recomendación X.500.
- La Recomendación X.500 proporciona una flexibilidad importante al permitir la definición de nuevos atributos y tiempo de ejecución X.500.
- Es comprensible que se utilice la inversión existente en la Recomendación X.500 en vez de tratar de crear una infraestructura completamente nueva.
- Permite al intermediario utilizar la infraestructura general X.500 para buscar direcciones de presentación de intermediarios y clientes que están vinculados, y utilizar las características de seguridad de la Recomendación X.500 para autenticar a los usuarios.

NOTA – Los detalles de cómo proporcionar la infraestructura y las características de seguridad de la Recomendación X.500 para la función intermediación están fuera del ámbito de la presente Especificación.

No es posible realizar completamente un intermediario de ODP utilizando la Recomendación X.500 porque hay diferencias importantes en el modelo operacional utilizado por el intermediario de ODP, entre las que cabe citar:

- Las operaciones del intermediario no corresponden directamente a las operaciones X.500.
- Las operaciones distribuidas se realizan utilizando información almacenada en enlaces de intermediario y atributos de intermediario, cuyo significado difiere de la distribución sustentada en la Recomendación X.500.

6 Esquema

El esquema X.500 describe la porción de un DIT X.500 utilizado para almacenar la información conocida por un intermediario. El esquema se basa en el modelo del directorio X.500 y se proporciona en el anexo A.

6.1 Generalidades

La información conocida por un intermediario determinado (el objeto información de intermediario) se mantiene en un subárbol del DIT X.500. Este subárbol se puede insertar en cualquier parte del DIT global y no se definen reglas de estructuras para controlar su posición. Cabe esperar que el subárbol del intermediario esté unido comúnmente por debajo de los asientos de organizaciones y de unidades de organización (que representan, respectivamente, la información conocida por los intermediarios de organizaciones y de unidades de organización). La información conocida por cada intermediario se mantiene separadamente en el DIT y no se intenta hacer corresponder el modelo de distribución X.500 con el modelo de distribución muy diferente de los intermediarios federados.

La información de intermediario se almacena en el DIT X.500 como parcelas independientes. Cada parcela contiene la información conocida por un intermediario. En el ejemplo mostrado en la figura 2, hay dos intermediarios: uno para toda la organización y un segundo para una unidad dentro de la organización. Las vinculaciones entre estos dos intermediarios se efectúan a través del protocolo de intermediario, no a través del protocolo X.500.

El objeto información de intermediario (véase la figura 3) se compone de cinco tipos de asientos:

- El asiento de intermediario, que contiene detalles sobre el propio intermediario.
- El asiento de política de intermediario, que contiene detalles sobre las políticas de empresa del intermediario.
- Los asientos de ofertas de servicio, que contienen detalles sobre las ofertas de servicio conocidas por el intermediario.
- Los asientos de enlace de intermediarios, que contienen detalles sobre los enlaces con otros intermediarios.
- Los asientos de ofertas por procuración, que contienen detalles sobre las ofertas por procuración conocidas por el intermediario.

NOTA 1 – La estructuración de las ofertas de servicio, de los enlaces y de las ofertas por procuración mostradas en la figura 3 es sólo un ejemplo de una posible estructura de información.

NOTA 2 – Además de los atributos X.500 enumerados en cada asiento, la presencia de otros atributos en un asiento no es una violación de esta especificación. Se pueden necesitar otros atributos X.500 por los siguientes motivos:

- si una determinada aplicación de intermediario requiere atributos X.500 adicionales específicos, éstos pueden ser definidos en esa especificación de aplicación de intermediario;
- si una determinada realización de intermediario requiere otros atributos X.500 adicionales, éstos pueden ser definidos en la documentación para esa realización.

Los atributos adicionales pueden ser incluidos como clases de objetos auxiliares.

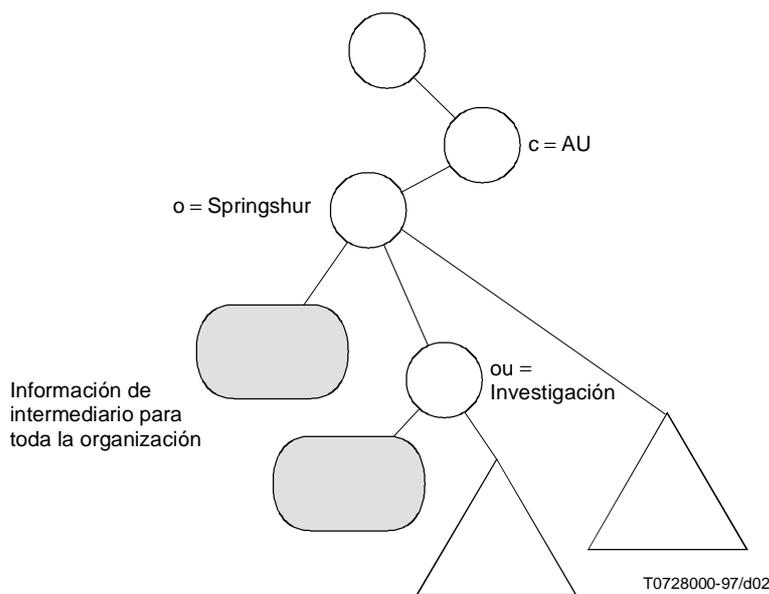


Figura 2 – Ejemplo de dos intermediarios almacenados en el DIT X.500

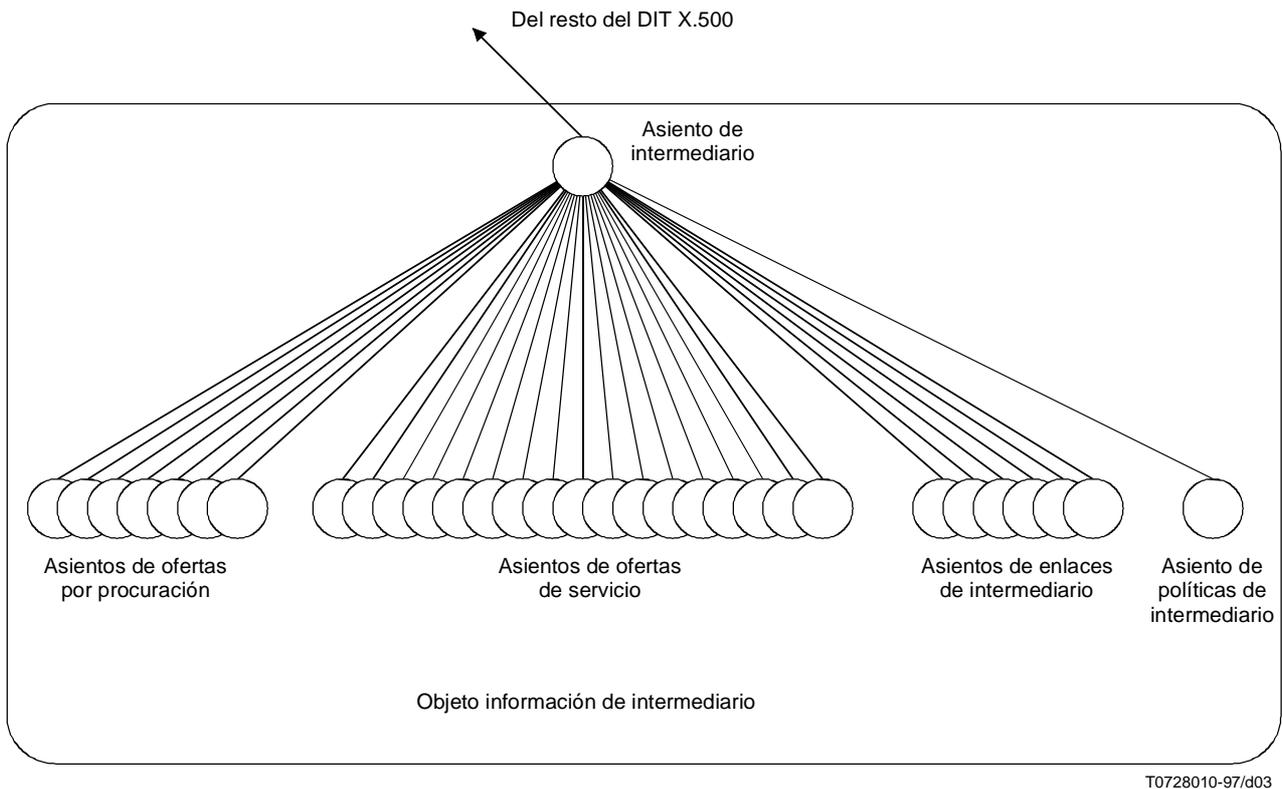


Figura 3 – Ejemplo de un objeto información de intermediario con cinco tipos de inserciones X.500

6.2 Asiento de intermediario (trader Entry)

La raíz del subárbol de intermediario es el asiento de intermediario, que contiene información sobre el propio intermediario (los atributos de intermediario – las características de intermediario normalizadas y las políticas de intermediación) y es utilizado como información de configuración por el intermediario (T-DUA) cuando actúa. La información se expresa como un conjunto de atributos X.500 que representan atributos de intermediario.

```
traderEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    MUST CONTAIN                {commonName | traderInterface | dsaName | typeRepos |
                                defSearchCard | maxSearchCard | defMatchCard |
                                maxMatchCard | defReturnCard | maxReturnCard |
                                defHopCount | maxHopCount | defFollowPolicy |
                                maxFollowPolicy | maxLinkFollowPolicy |
                                supportsModifiableProperties | supportsDynamicProperties |
                                supportsProxyOffers | maxList | requestIdStem}
    MAY CONTAIN                  {description | userPassword}
    ID                           id-trader-oc-traderEntry}
```

6.2.1 Nombre común (commonName)

El nombre de este intermediario. El atributo `commonName` forma el RDN del asiento de intermediario. El nombre completo de un intermediario es el nombre distinguido de este asiento (es decir, el "nombre de trayecto" completo del asiento de intermediario en el DIT X.500 global). El nombre distinguido completo identifica inequívocamente a este intermediario entre todos los otros intermediarios en el directorio X.500. Éste es un atributo X.500 normalizado definido en la Rec. UIT-T X.520 | ISO/CEI 9594-6.

6.2.2 Interfaz de intermediario (traderInterface)

La dirección del intermediario. La "dirección" es la dirección de presentación en la cual es posible ponerse en contacto con este intermediario. Este atributo X.500 es utilizado por el intermediario cuando funciona como parte de su información de configuración y también por otros intermediarios cuando desean distribuir una importación de intermediario entre intermediarios federados.

```
traderInterface ATTRIBUTE ::= {
    SUBTYPE OF          presentationAddress
    SINGLE VALUE        TRUE
    ID                  id-trader-at-traderInterface}
```

6.2.3 Nombre de DSA (dsaName)

El nombre para el DSA asociado con el objeto intermediario.

```
dsaName ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-dsaName}
```

6.2.4 Depositario de tipos (typeRepos)

El nombre del depositario de tipos utilizado por el intermediario para el depósito de definiciones de tipos de servicios, tipos de interfaz y tipos de propiedades de servicio.

```
typeRepos ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-typeRepos}
```

6.2.5 Tarjeta de búsqueda por defecto (defSearchCard)

El límite superior por defecto de las ofertas de servicio que ha de ser considerado antes de terminar una búsqueda. Este valor se utiliza si ningún valor es especificado por un importador. No debe exceder del valor de maxSearchCard.

```
defSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defSearchCard }
```

6.2.6 Tarjeta de búsqueda máxima (maxSearchCard)

El límite superior máximo de ofertas de servicio que un intermediario considera antes de terminar cualquier búsqueda.

```
maxSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxSearchCard}
```

6.2.7 Tarjeta de concordancia por defecto (defMatchCard)

El límite superior por defecto de ofertas concordadas halladas antes de que un intermediario termine una búsqueda. Este valor se utiliza si ningún valor es especificado por un importador. No debe exceder del valor de maxMatchCard.

```
defMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defMatchCard}
```

6.2.8 Tarjeta de concordancia máxima (maxMatchCard)

El límite superior máximo de ofertas concordadas halladas antes de que un intermediario termine cualquier búsqueda.

```
maxMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxMatchCard}
```

6.2.9 Tarjeta de devolución por defecto (defReturnCard)

El límite superior por defecto de ofertas de servicio devueltas a un importador. Este valor se utiliza si ningún valor es especificado por un importador. No debe exceder del valor máximo de maxReturnCard.

```
defReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defReturnCard}
```

6.2.10 Tarjeta de devolución máxima (maxReturnCard)

El límite superior máximo de ofertas de servicios devueltas a un importador.

```
maxReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxReturnCard}
```

6.2.11 Cómputo de saltos por defecto (defHopCount)

El límite superior por defecto de la profundidad de enlaces que han de ser atravesados antes de terminar una búsqueda. Este valor se utiliza si ningún valor es especificado por un importador. No debe exceder de maxHopCount.

```
defHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-defHopCount }
```

6.2.12 Cómputo de saltos máximo (maxHopCount)

El límite superior máximo de la profundidad de enlaces que han de ser atravesados antes de terminar cualquier búsqueda.

```
maxHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-maxHopCount }
```

6.2.13 Política de seguimiento por defecto (defFollowPolicy)

El comportamiento de seguimiento de enlaces por defecto, cuando ningún comportamiento de seguimiento de enlaces es especificado por un importador. El comportamiento de seguimiento en un enlace puede ser uno de los siguientes:

- local solamente – no seguir nunca, a menos que se indique explícitamente en una operación;
- si no hay local – seguir solamente si ninguna oferta local puede servir;
- siempre – seguir siempre, salvo cuando está abrogado por algunas políticas.

```
defFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-defFollowPolicy }
```

```
FollowOption ::= ENUMERATED{
    localOnly    (0),
    ifNoLocal   (1),
    always      (2)}
```

6.2.14 Política de seguimiento máximo (maxFollowPolicy)

La política de seguimiento máximo en todos los enlaces en un intermediario para una indagación dada. Puede abrogar las políticas de enlace y de importador en el comportamiento de seguimiento de enlaces.

```
maxFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxFollowPolicy }
```

6.2.15 Política de seguimiento de enlace máximo (maxLinkFollowPolicy)

El comportamiento de seguimiento de enlaces más permisivo para un enlace cuando se crea o modifica el comportamiento limitador en cualquier enlace en un intermediario.

```
maxLinkFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxLinkFollowPolicy }
```

6.2.16 Admite propiedades modificables (supportsModifiableProperties)

Si es "verdadero", este atributo permite invocar la operación modificar oferta.

```
supportsModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsModifiableProperties }
```

6.2.17 Admite propiedades dinámicas (supportsDynamicProperties)

Si es "verdadero", este atributo permite que las ofertas de servicio tengan propiedades dinámicas. Es decir, propiedades cuyos valores tienen que ser obtenidos cuando lo requiera la concordancia o la descripción de ofertas de servicios y/o la concordancia de ofertas por procuración.

```
supportsDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsDynamicProperties }
```

6.2.18 Admite ofertas por procuración (supportsProxyOffers)

Si es "verdadero", este atributo permite exportar, retirar, describir y enumerar ofertas por procuración, que son ofertas de servicio especiales que proporcionan la determinación del tiempo de ejecución de la interfaz en la cual se proporciona el servicio anunciado.

```
supportsProxyOffers ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsProxyOffers }
```

6.2.19 Lista máxima (maxList)

El tamaño de lista máximo para un iterador que el intermediario desea apoyar.

```
maxList ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxList }
```

6.2.20 Petición de origen de identificador (requestIdStem)

Una identificación del intermediario, que se ha de utilizar como el origen para la producción de un identificador en una petición de indagación que es iniciada por este intermediario para ser transferida a un intermediario vinculado.

```
requestIdStem ATTRIBUTE ::= {
    WITH SYNTAX                OCTET STRING (SIZE (0..ub-request-id-stem))
    EQUALITY MATCHING RULE     octetStringMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-requestIdStem }
```

6.2.21 Descripción (description)

Una descripción textual del intermediario. La "descripción" es un campo de texto libre que describe al intermediario (por ejemplo "CSIRO Intermediario de la División de Tecnología de la Información"). Éste es un atributo X.500 normalizado definido en la Rec. UIT-T X.520 | ISO/CEI 9594-6.

6.2.22 Contraseña de usuario (userPassword)

Una contraseña para proporcionar autenticación simple cuando el T-DUA accede al objeto información de intermediario. Las reglas de control de acceso para este atributo no deben permitir que este atributo sea legible generalmente. Éste es un atributo X.500 normalizado definido en la Rec. UIT-T X.509 | ISO/CEI 9594-8.

6.2.23 Otros atributos X.500

Otros atributos X.500 requeridos por una realización o por una aplicación específicas pueden ser incluidos como clases de objetos auxiliares. Como ejemplos de posibles atributos X.500 adicionales en este asiento cabe citar:

- Información de control de acceso (en el asiento de intermediario).
- Información de contacto para el administrador del intermediario.
- Información de contacto humano para el servicio, que incluye una descripción textual del servicio.
- Información límite (por ejemplo, la cantidad máxima de recursos que han de ser consumidos por una indagación, la duración máxima de una oferta).

6.3 Asiento de políticas de intermediario (traderPolicyEntry)

El asiento de políticas de intermediario está situado inmediatamente debajo del asiento de intermediario en el subárbol de información. Contiene las políticas de empresa del intermediario (políticas definidas en la especificación de empresa de la función intermediario en la Rec. UIT-T X.950 | ISO/CEI 13235-1), expresadas como un conjunto de constricciones de política. Cada restricción de política se puede expresar como una cadena que describe la regla de política que consiste en un filtro X.500 compuesto de atributos del asiento de intermediario, o el nombre de un objeto que aplica esa política.

```
traderPolicyEntryNF NAME-FORM ::= {
    NAMES                traderPolicyEntry
    WITH ATTRIBUTES      {commonName}
    ID                   id-trader-nf-traderPolicy}

traderPolicyEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {commonName }
    MAY CONTAIN          {typeManagementConstraint | searchConstraint |
                        offerAcceptanceConstraint }
    ID                   id-trader-oc-traderPolicy}
```

Las constricciones de políticas se definen como:

```
PolicySpecification ::= CHOICE {
    stringRule           [0]   DirectoryString{ub-policy-string-rule}
    policyObjectId       [1]   DistinguishedName }
```

```
policySpecificationMatch MATCHING-RULE ::= {
    SYNTAX               PolicySpecification
    ID                   id-trader-mr-policySpecificationMatch}
```

-- The rule returns TRUE if two Specifications contain exactly the same characters.

6.3.1 Nombre común (commonName)

El atributo commonName forma el RDN del asiento de políticas de intermediario y tiene el valor "políticas de intermediario". Éste es un atributo X.500 normalizado definido en la Rec. UIT-T X.520 | ISO/CEI 9594-6.

6.3.2 Constricción de gestión de tipos (typeManagementConstraint)

Esta restricción es una regla relacionada con la especificación de tipos y la relación entre tipos.

```
typeManagementConstraint ATTRIBUTE ::= {
    WITH SYNTAX          PolicySpecification
    EQUALITY MATCHING RULE policySpecificationMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-typeManagementConstraint}
```

6.3.3 Constricción de búsqueda (searchConstraint)

Esta constricción es una regla que orienta la búsqueda de ofertas adecuadas a través del sistema de intermediario, por ejemplo, búsqueda secuencial o búsqueda paralela de intermediarios federados.

```
searchConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-searchConstraint}
```

6.3.4 Constricción de aceptación de oferta (offerAcceptanceConstraint)

Esta constricción es una regla que restringe el conjunto de ofertas de servicio aceptables para el intermediario. Por ejemplo, un intermediario puede sólo aceptar ofertas de servicio con tipos de servicio específicos.

```
offerAcceptanceConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-offerAcceptanceConstraint}
```

6.3.5 Otros atributos X.500

Otros atributos X.500 requeridos por una realización o por una aplicación específicas pueden ser incluidos como clases de objetos auxiliares. Como ejemplos de posibles atributos X.500 adicionales en este asiento cabe citar:

- Información de control de acceso (en el asiento de políticas intermediario).
- Constricciones adicionales de políticas de empresa del intermediario.

6.4 Asiento de oferta de servicios (serviceOfferEntry)

Los asientos de ninguna o más ofertas de servicio están situados inmediatamente debajo del asiento de intermediario en el subárbol de información. Cada asiento de oferta de servicio contiene los detalles de una oferta. Colectivamente, los asientos de ofertas de servicio forman el conjunto de ofertas de servicio. Cada asiento de oferta de servicio tiene una clase de objeto estructural de asiento de oferta de servicio y exactamente una clase de objeto auxiliar de las enumeradas en la regla de contenido para el asiento de oferta de servicio. Cada clase de objeto auxiliar corresponde a un tipo de servicio y define las propiedades de servicio que son obligatorias o facultativas. Por consiguiente, cada asiento de oferta de servicio contiene los atributos obligatorios y facultativos del asiento de oferta de servicio, y los atributos obligatorios y facultativos determinados por la correspondiente clase de objeto auxiliar, que especifica las propiedades del servicio que son obligatorias y facultativas. Cada asiento de oferta de servicio es denominado utilizando el atributo identificador de oferta de servicio de el asiento.

```
serviceOfferEntryNF NAME-FORM ::= {
    NAMES                        serviceOfferEntry
    WITH ATTRIBUTES              {sOfferId}
    ID                            id-trader-nf-serviceOffer}
serviceOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF                  {top}
    MUST CONTAIN                 {sOfferId | serviceInterfaceId | serviceTypeId
                                hasDynamicProperties | hasModifiableProperties }
    MAY CONTAIN                   {dynamicProps}
    ID                            id-trader-oc-serviceOffer}
```

Las propiedades de una oferta de servicio se almacenan como atributos X.500. Los atributos particulares (propiedades obligatorias del servicio) que deben estar presentes en una oferta de servicio y las que son permitidas (propiedades facultativas del servicio) en una oferta de servicios son controladas por el esquema X.500. El nombre de propiedad corresponde a un tipo de atributo X.500 y los valores de propiedad corresponden a valores de atributos X.500. Los nombres de propiedades son identificadores de objeto. Las propiedades del servicio son controladas por la clase de objeto auxiliar asociada con el tipo de servicio de la oferta de servicio. Además, una propiedad de una oferta de servicio puede ser controlada por las reglas de control de acceso, para que no sean modificables por un usuario (es decir, lectura solamente).

Cada identificador de tipo de servicio tiene asociados un tipo de interfaz y un tipo de propiedades del servicio que pueden ser obtenidos del depositario de tipos. El exportador debe especificar un identificador de tipo de servicio válido conocido por el intermediario. El identificador de tipo de propiedades de servicio asociado se representa como una clase de objeto auxiliar X.500 y en consecuencia es almacenado en cada asiento de oferta de servicio como un valor en el atributo clase de objeto. Esta clase de objeto auxiliar define los atributos X.500 obligatorios y facultativos (propiedades del servicio) que deben/pueden estar contenidos en una oferta de este tipo de servicio.

Las clases de objeto auxiliares pueden ser derivadas de otras clases de objetos X.500 (denominadas "subclases" en la Recomendación X.500). Esto equivale a derivar la definición de un tipo de propiedades de servicio a partir de otro. Las reglas para las subclases pueden comprenderse mejor en relación con las listas de lo que deben y pueden contener:

- La subclase debe contener cualesquiera atributos de la "lista de lo que debe contener" de su superclase.
- La subclase puede contener cualesquiera atributos de la "lista de lo que puede contener" de su superclase.
- Una clase de objeto puede ser una subclase de dos o más clases. La subclase debe contener todos los atributos de todas las "listas de lo que debe contener" de todas sus superclases y "puede contener" cualquiera de los atributos de sus superclases.

NOTA – En la definición de asientos de ofertas de servicio, la clase de objeto estructural de un asiento de servicio es la oferta de "servicios genéricos", mientras que el "tipo de propiedades de servicio" que especializa la oferta de servicio a un tipo de servicio particular es una clase de objeto auxiliar. Habría sido posible que la "oferta de servicio" fuese una clase de objeto abstracta, y entonces los diversos "tipos de propiedades de servicio" habrían sido clases de objeto estructurales derivadas de la clase abstracta "oferta de servicio". Esto tendría la desventaja de requerir un esquema X.500 complejo con muchas reglas.

Para una oferta dada, un exportador puede también decir que una propiedad particular tiene un valor dinámico. Es decir, el valor de la propiedad no se especifica en el momento de la exportación pero ha de ser evaluado cuando se requiera al comparar o describir las ofertas de servicio (por ejemplo, mediante la utilización de las operaciones de intermediario indagar o describir). En cambio, el exportador especifica la interfaz en la cual la propiedad puede ser evaluada.

6.4.1 Identificador de oferta de servicio (sOfferId)

El sOfferId es el identificador del asiento de oferta de servicio que es asignado por el intermediario. Forma el RDN del asiento de oferta de servicio. El identificador se elige para que sea único entre todos los identificadores de ofertas de servicio mantenidos por ese intermediario. El sOfferId forma parte del identificador de oferta de servicio que el T-DUA devuelve al exportador de la oferta de servicio después del procesamiento satisfactorio de una petición de la operación exportar.

NOTA 1 – El T-DUA puede elegir nombres utilizando cualquier algoritmo, a condición de que los identificadores resultantes sean únicos. Un contador por incrementos es suficiente.

```
sOfferId ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString{ub-s-offer-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                         id-trader-at-sOfferId}
```

NOTA 2 – El exportador recibe un identificador de oferta de servicio que es la representación en una cadena del nombre de directorio parcial (bajo la raíz del intermediario) del asiento de oferta de servicio utilizando la notación de valor ASN.1. Para el ejemplo de la figura 3, un identificador de oferta de servicio devuelto al exportador podría ser {sOfferId, 5}.

6.4.2 Identificador de interfaz de servicio (serviceInterfaceId)

La interfaz de servicio contiene el nombre de un asiento de interfaz (véase 6.7) que contiene a su vez información de la interfaz del servicio exportado. El intermediario utiliza este nombre para leer el asiento de interfaz con el fin de extraer información sobre esta interfaz (por ejemplo, la dirección de presentación o la referencia de interfaz de ingeniería).

```
serviceInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-serviceInterfaceId}
```

NOTA – Cuando se trata solamente con una infraestructura, una alternativa es sustituir la especificación anterior serviceInterfaceId por:

```
ServiceIntId ::= SEQUENCE {
    serviceInterfaceType  DirectoryString{ub-service-interface-type}
    interfaceRef          OPEN TYPE}

serviceInterfaceId ATTRIBUTE ::= {
    WITH SYNTAX          ServiceIntId
    EQUALITY MATCHING RULE  caseExactMatch
    ID                  id-trader-at-serviceInterfaceId}
```

6.4.3 Identificador de tipo de servicio (serviceTypeId)

Es el identificador del tipo de servicio que está en oferta. El identificador de tipo de servicio controla el tipo de interfaz y el tipo de propiedades de servicio que pueden estar asociados con esta oferta de servicio.

```
serviceTypeId ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE  objectIdentifierMatch
    ID                  id-trader-at-serviceTypeId }
```

6.4.4 Tiene propiedades dinámicas (hasDynamicProperties)

Si está puesto a "verdadero", este atributo indica que esta oferta de servicio contiene propiedades dinámicas. Es decir, propiedades cuyos valores tienen que ser obtenidos cuando se requiera para concordar o describir ofertas de servicio. En su política de importación, un importador puede pedir que no se consideren ofertas con propiedades dinámicas.

```
hasDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE  booleanMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-hasDynamicProperties}
```

6.4.5 Tiene propiedades modificables (hasModifiableProperties)

Si está puesto a "verdadero", este atributo indica que esta oferta de servicio contiene propiedades que no son de lectura solamente. Es decir, propiedades cuyos valores pueden ser cambiados cuando lo requiera el exportador. En su política de importación, un importador puede pedir que no se consideren ofertas con propiedades modificables.

```
hasModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                               BOOLEAN
    EQUALITY MATCHING RULE                    booleanMatch
    SINGLE VALUE                              TRUE
    ID                                         id-trader-at-hasModifiableProperties}
```

6.4.6 Propiedades dinámicas (dynamicProps)

Este atributo se fija cuando una oferta de servicio contiene propiedades dinámicas, que son propiedades cuyos valores se han de obtener cuando se necesiten para concordar o describir ofertas de servicio. Cada exportador puede determinar qué propiedades (obligatorias o facultativas) son dinámicas. Sin embargo, si ha de ser determinada para evaluación dinámica, la propiedad no debe ser una propiedad de lectura solamente. Este atributo contiene la información requerida para obtener los valores para una o más propiedades dinámicas. Se fija un valor de propiedad dinámica para cada propiedad dinámica en una oferta de servicio.

```
dynamicProps ATTRIBUTE ::= {
    WITH SYNTAX                               SEQUENCE OF DynamicPropValue
    ID                                         id-trader-at-dynamicProps}

dynamicPropValue ::= SEQUENCE {
    propertyType                             OBJECT IDENTIFIER,
    dynamicPropEvalIf                        DistinguishedName,
    extraInfo                                DirectoryString{ub-dynamic-value-extra-info}}

DynamicPropValueMatch MATCHING-RULE ::= (
    SYNTAX                                    DynamicPropValue,
    ID                                        id-trader-mr-dynamicPropValueMatch}

-- The rule returns TRUE if two values contain exactly the same characters.
```

6.4.7 Otros atributos X.500

Además de los atributos ya enumerados, un asiento de oferta de servicio normalmente contiene otros atributos que representan las propiedades del servicio. Los atributos de propiedades del servicio son específicos del tipo de servicio y son controlados por una clase de objeto auxiliar. El tipo de propiedades del servicio especifica si una propiedad es obligatoria o facultativa, y (con información de control de acceso) si un usuario puede modificar su valor. El atributo propiedades dinámicas es específico de una oferta particular, por ejemplo, una oferta de un servicio determinado de un exportador puede no tener valores dinámicos mientras que una oferta de servicio diferente del mismo tipo de servicio puede contener varias propiedades dinámicas.

Otros atributos adicionales X.500 requeridos por una realización o por una aplicación específicas pueden ser incluidos como clases de objetos auxiliares. Como ejemplos de posibles atributos X.500 adicionales en este asiento cabe citar:

- Información de control de acceso (en el asiento de oferta de servicio).
- Fecha de expiración de una oferta de servicio.
- Información de contacto humano para el servicio, incluida una descripción textual del servicio.

6.5 Asiento de enlace de intermediarios (traderLinkEntry)

Ninguno o más asientos de enlace de intermediarios están situados inmediatamente debajo del asiento de intermediario en el subárbol de información. Cada asiento de enlace de intermediario contiene los detalles de un enlace de intermediarios. Colectivamente, los asientos de enlace de intermediarios forman el conjunto de enlaces de intermediarios. Cada asiento de enlace de intermediarios es denominado utilizando el atributo identificador de enlace de este asiento.

```
traderLinkEntryNF NAME-FORM ::= {
    NAMES                traderLinkEntry
    WITH ATTRIBUTES      {linkId}
    ID                   id-trader-nf-traderLink}
```

```
traderLinkEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {linkName | linkId | targetTraderInterfaceId |
                        defPassOnFollowRule | limitingFollowRule}
    ID                   id-trader-oc-traderLink}
```

6.5.1 Nombre de enlace (linkName)

Es el nombre del enlace suministrado por el administrador del intermediario de origen cuando crea el enlace. El nombre debe ser único entre todos los nombres de enlace mantenidos por el intermediario.

```
linkName ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-link-name}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-linkName}
```

6.5.2 Identificador de enlace (linkId)

Es el identificador de enlace asignado por el intermediario. Forma el RDN de un asiento de enlace de intermediarios. El identificador debe ser único entre todos los identificadores de enlace de intermediarios mantenido por ese intermediario.

NOTA – El T-DUA puede elegir nombres utilizando cualquier algoritmo, a condición de que los identificadores resultantes sean únicos. Un contador por incrementos es suficiente.

```
linkId ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-link-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-linkId}
```

6.5.3 Identificador de interfaz de intermediario deseado (targetTraderInterfaceId)

Este atributo contiene el nombre de la interfaz del intermediario deseado señalado por el enlace. El nombre se utiliza para buscar la dirección de presentación del intermediario distante.

```
targetTraderInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-targetTraderInterfaceId}
```

6.5.4 Regla de seguimiento de transferencia por defecto (defPassOnFollowRule)

El comportamiento de seguimiento de enlaces por defecto en un enlace se especifica cuando se crea el enlace. No debe rebasar la regla limitadora de seguimiento para este enlace.

```
defPassOnFollowRule ATTRIBUTE ::= {
    WITH SYNTAX          FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defPassOnFollowRule }
```

6.5.5 Regla de limitadora de seguimiento (limitingFollowRule)

El comportamiento de seguimiento de enlace límite de un determinado enlace se especifica cuando se crea el enlace. Su valor no debe rebasar la política de seguimiento de enlace máximo del intermediario de origen en el momento de la creación del enlace. Sin embargo, es admisible si un cambio ulterior de esta política hace que la regla limitadora de seguimiento exceda del valor de la política de seguimiento de enlace máximo modificada del intermediario.

```
limitingFollowRule ATTRIBUTE ::= {
    WITH SYNTAX          FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-limitingFollowRule }
```

6.5.6 Otros atributos X.500

Además de los atributos X.500 ya enumerados, un asiento de enlace de intermediario contiene normalmente otros atributos que representan información del intermediario deseado percibido por el intermediario de origen. Estos atributos de enlace son utilizados por el intermediario de origen para ayudar a decidir qué enlace será el siguiente. Estos atributos X.500 y otros atributos X.500 adicionales requeridos por una realización o aplicación específicas pueden ser incluidos como clases de objetos auxiliares. Como ejemplo de posibles atributos X.500 adicionales en este asiento cabe citar:

- Información de control de acceso (en el asiento de enlace de intermediarios).
- Atributos sobre el asiento de intermediarios (distantes) vinculados, incluida la información de contacto humano para el intermediario vinculado y una descripción textual del intermediario vinculado.
- El periodo durante el cual este enlace es válido.
- La última vez que se invocó el enlace.

6.6 Asiento de oferta por procuración (proxyOfferEntry)

Los asientos de ninguna o más ofertas por procuración están situados inmediatamente debajo del asiento de intermediario en el subárbol de información. No tiene que haber ningún asiento de oferta por procuración si el atributo de intermediario "admite ofertas por procuración" está puesto a "falso". Si este atributo del asiento de intermediario está puesto a "verdadero", entonces cada asiento de oferta por procuración contiene los detalles de una oferta por procuración. Colectivamente, los asientos de oferta por procuración forman el conjunto de ofertas por procuración. Cada asiento de oferta por procuración tiene una clase de objeto estructural de asiento de oferta por procuración y exactamente una clase de objeto auxiliar de las enumeradas en la regla de contenido para el asiento de oferta por procuración. Cada clase de objeto auxiliar corresponde a un tipo de propiedades de servicio y define las propiedades de servicio que son obligatorias o facultativas. Por consiguiente, cada asiento de oferta por procuración contiene los atributos obligatorios y facultativos del asiento de oferta por procuración, y los atributos obligatorios y facultativos determinados por la clase de objeto auxiliar correspondiente, que especifica las propiedades de servicio obligatorias y facultativas. Cada asiento de oferta por procuración es denominada utilizando el atributo identificación de oferta por procuración del asiento.

```

proxyOfferEntryNF NAME-FORM ::= {
    NAMES                proxyOfferEntry
    WITH ATTRIBUTES      {proxyOfferId}
    ID                   id-trader-nf-proxyOffer}

proxyOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {proxyOfferId | proxyLookUpInterfaceId |
                        hasDynamicProperties | hasModifiableProperties |
                        ifMatchAll | constraintRecipe}
    MAY CONTAIN          { dynamicProps }
    ID                   id-trader-oc-proxyOffer}

```

Una oferta por procuración es un cruce entre una oferta de servicio y una forma de enlace restringido. Como las ofertas de servicio normales, las ofertas por procuración tienen un tipo de servicio y un conjunto de propiedades de servicio. Sin embargo, una oferta por procuración no incluye una interfaz en la cual se proporciona el servicio ofrecido. En cambio, proporciona una interfaz de consulta, a partir de la cual se invoca una operación indagar modificada y a partir de la cual se obtiene una determinación del tiempo de ejecución de la interfaz para proporcionar el servicio.

Además, una oferta por procuración tiene una constricción (utilizada para formular una constricción modificada en la indagación modificada). Una oferta por procuración puede tener también una secuencia de pares de nombre-valor que se ha de añadir a la política de importador original para formar la política de importador en la operación indagar modificada.

6.6.1 Identificador de oferta por procuración (proxyOfferId)

Es el RDN del asiento y se elige de modo que sea único entre todos los nombres de ofertas por procuración mantenidos por ese intermediario. El identificador de oferta por procuración forma parte del identificador de oferta por procuración que el T-DUA devuelve al exportador de la oferta por procuración después del procesamiento satisfactorio de una petición de la operación exportar por procuración.

NOTA 1 – El T-DUA puede elegir cualesquiera nombres utilizando cualquier algoritmo, a condición de que los nombres resultantes sean únicos. Un contador por incrementos es suficiente.

NOTA 2 – El exportador recibe un identificador de oferta por procuración que es la representación en cadena del nombre de directorio parcial (bajo la raíz del intermediario) del asiento de oferta por procuración utilizando la notación de valor ASN.1.

```

proxyOfferId ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-proxy-offer-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-proxyOfferId}

```

6.6.2 Identificador de interfaz de consulta por procuración (proxyLookupInterfaceId)

Este identificador es el nombre de un asiento que representa una interfaz que puede aceptar una operación indagar del intermediario. El intermediario utiliza este nombre para leer el asiento con el fin de extraer información (por ejemplo, la dirección de presentación) sobre esta interfaz.

```
proxyLookupInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyLookupInterfaceId}
```

6.6.3 Fórmula de restricción (constraintRecipe)

Este atributo proporciona el modo de convertir una restricción de concordancia del importador de la petición de indagación original en una nueva restricción de concordancia que se ha de utilizar en la petición de indagación modificada en la interfaz de consulta por procuración. El lenguaje de restricción se especifica en el anexo C a la Rec. UIT-T X.950 | ISO/CEI 13235-1.

```
constraintRecipe ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-constraint-recipe}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-constraintRecipe}
```

6.6.4 Si todo concuerda (ifMatchAll)

Si este atributo está puesto a "verdadero", el intermediario debe concordar esta oferta por procuración con una indagación de un importador sobre el tipo de servicio solamente, es decir, no es necesario comparar la expresión de "restricción" del importador contra las propiedades asociadas con esta oferta por procuración para una búsqueda.

```
ifMatchAll ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE booleanMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-ifMatchAll }
```

6.6.5 Otros atributos X.500

Además de los atributos ya enumerados, un asiento de oferta por procuración, al igual que un asiento de oferta de servicio, contiene normalmente otros atributos que representan las propiedades del servicio, y si hay propiedades dinámicas y propiedades modificables. Los atributos de propiedades del servicio son específicos del tipo de servicio y son controlados por una clase de objeto auxiliar. Un tipo de atributo de propiedad especifica si un atributo es obligatorio o facultativo y (con información del control de acceso) si un usuario puede modificar su valor. Si hay propiedades dinámicas, se almacenan en el atributo propiedades dinámicas (véase 6.4.6).

La especificación de la función intermediación en la Rec. UIT-T X.950 | ISO/CEI 13235-1 permite también que se incluyan pares de nombre-valor adicionales en una oferta por procuración. Estos pares de nombre-valor, si están presentes, se han adjuntar al parámetro de política de la petición de indagación original, para transferir estos pares de nombre-valor a una fábrica de servicios para su utilización durante la creación del servicio por la fábrica de servicios. Los pares de nombre-valor son representados como pares de tipo de atributo y de valor de atributo X.500.

Se pueden incluir otros atributos X.500 adicionales requeridos por una realización o aplicación específica. Como ejemplos de posibles atributos X.500 en este asiento cabe citar:

- Información de control de acceso (sobre el asiento de oferta por procuración).
- Información de contacto humano para la oferta por procuración, incluida una descripción textual del servicio.

6.7 Otros asientos X.500 utilizados por el T-DUA

Además de los cinco tipos de asientos en el objeto información de intermediario, el T-DUA puede también utilizar otros asientos en la base de datos X.500, que incluyen asientos que representan:

- Individuos, unidades de organización, u organizaciones que pueden almacenar información de autenticación.
- Interfaces, que contienen información sobre la interfaz, por ejemplo, la dirección (o una referencia de interfaz de ingeniería) utilizada para acceder a esa interfaz. La especificación de la función intermediario (Rec. UIT-T X.950 | ISO/CEI 13235-1) especifica que hay que indicar una referencia de interfaz de ingeniería para cada infraestructura. Con el fin de mantener la flexibilidad máxima, cada interfaz de la especificación computacional es representada por un asiento de directorio distinto. El asiento de directorio puede tener atributos adaptados para cada infraestructura, según proceda. Esto añade un nivel de dificultad para acceder a las direcciones de interfaz; el intermediario almacena el nombre de un asiento que contiene información de interfaz y se accede a ese asiento para extraer la dirección. Una ventaja es que si cambia la dirección en la cual se accede a la interfaz, sólo hay que cambiar un asiento, en vez de todas las referencias a esa interfaz en todos los intermediarios. Esto es particularmente importante cuando la interfaz tiene que ser conocida por muchos intermediarios, por ejemplo, un depositario de tipos.

Los asientos de interfaz contienen un atributo de referencia de interfaz (interfaceReference) y un atributo de tipo de interfaz (interfaceType). El identificador de objeto para el tipo de interfaz debe ser conocido por el depositario de tipos.

```
interfaceEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    KIND                        auxiliary
    MUST CONTAIN                {interfaceReference | interfaceType}
    ID                          id-trader-oc-interfaceEntry}

interfaceReference ATTRIBUTE ::= {
    WITH SYNTAX                 InterfaceId
    EQUALITY MATCHING RULE     caseExactMatch
    ID                          id-trader-at-interfaceReference}

InterfaceId ::=                DirectoryString {ub-interface-id}

interfaceType ATTRIBUTE ::= {
    WITH SYNTAX                 InterfaceTypeName
    EQUALITY MATCHING RULE     objectIdentifierMatch
    ID                          id-trader-at-interfaceType}
InterfaceTypeName ::=         OBJECT IDENTIFIER
```

7 Operaciones

Esta cláusula describe la correspondencia entre las funciones de intermediario especificadas en la Rec. UIT-T X.950 | ISO/CEI 13235-1 y las operaciones que el T-DUA ejecuta en el directorio X.500 para realizar estas funciones. Se supone que los exportadores e importadores comunican con el T-DUA a través de un protocolo de intermediario que admite la funcionalidad requerida en la Rec. UIT-T X.950 | ISO/CEI 13235-1.

7.1 Inicialización

Cuando se establece un nuevo intermediario, el asiento de intermediario es creado por el administrador (directamente o a través de un programa que lee información de configuración). El nuevo asiento contiene la información de configuración para el nuevo intermediario.

Cuando el intermediario (T-DUA) comienza, se vincula al directorio y lee el asiento de intermediario para descubrir la información sobre sí mismo. La única información necesaria para el T-DUA es:

- El nombre distinguido de su asiento de intermediario.
- Una contraseña.

El T-DUA se vincula al DSA X.500 cuando el intermediario es creado y permanece vinculado hasta que el intermediario es destruido (salvo en el caso de colisiones del intermediario o del sistema X.500). El intermediario debe vincularse al DSA con privilegios suficientes para modificar el asiento de intermediario, y crear, leer, suprimir y modificar los asientos de ofertas de servicio, los asientos de enlace de intermediarios y, si se admite la oferta por procuración, crear, leer y suprimir asientos de oferta por procuración.

7.2 Operaciones de clientes

Un cliente utiliza un protocolo de intermediario para comunicar con el T-DUA. Este protocolo no se define detalladamente en la presente Recomendación. La operación solicitada por el cliente origina una o más operaciones en la base de datos X.500. La correspondencia entre las operaciones del intermediario y las operaciones X.500 se describen en las secciones siguientes. La correspondencia se produce solamente después que el T-DUA ha comprobado la sintaxis de cada parámetro según se especifica en el anexo A a la Rec. UIT-T X.950 | ISO/CEI 13235-1; un parámetro incorrecto sintácticamente es devuelto como una excepción con el prefijo "ilegal" y la operación no es procesada.

Varias operaciones de intermediario requieren cierta forma de seguridad (por ejemplo, sólo un cliente que actúa como administrador puede añadir o suprimir enlaces). Esta seguridad es aplicada aprovechando la seguridad proporcionada por la Recomendación X.500. Para ejecutar una operación de intermediario, un cliente debe suministrar un nombre y una contraseña. En esta especificación, se supone que éstos sean un nombre de usuario y una contraseña de usuario X.500, o que correspondan directamente con éstos, posiblemente protegidos en alguna forma. La forma en que esto se hace depende del protocolo de intermediario utilizado para vincular un cliente a un intermediario.

El T-DUA utiliza esta autenticación para vincularse al DSA X.500 como ese usuario. La Recomendación X.500 proporciona facilidades para proteger la contraseña cuando se envía a través de la red entre el T-DUA y el DSA X.500. Hay una implicación de seguridad en este método: el cliente debe confiar en el T-DUA, pues el T-DUA puede ver la contraseña del cliente. El protocolo X.500 proporciona mecanismos para proteger contraseñas cuando son transmitidas entre los DUA y los DSA. Debe ser factible extender esta protección para que funcione entre un cliente y el DSA, puesto que el T-DUA no tiene que conocer la contraseña.

Una vez vinculado como el cliente al DSA, el T-DUA utiliza las características de seguridad X.500 para controlar el acceso a asientos en el árbol de información del intermediario. En la Rec. UIT-T X.950 | ISO/CEI 13235-1 no se define ninguna excepción de control de acceso específica, y se devuelve una excepción específica del sistema cuando no se autoriza el acceso [por ejemplo, en CORBA, se aplica la excepción normalizada "No hay permiso" (No_Permission)].

7.3 Operaciones de registro

Las operaciones de registro forman un conjunto de operaciones que permiten a un cliente exportar, retirar, modificar y describir ofertas de servicio en un intermediario. Además, hay una operación para resolver un nombre relativo de contexto de un intermediario.

7.3.1 Exportar

El intermediario recibe una operación exportar. Esta operación corresponde a una operación X.500 añadir asiento, para agregar un nuevo asiento de oferta de servicio al directorio X.500. Los datos en la operación exportar son:

Parámetro	Sintaxis de atributos X.500	Comentario
reference	Name	El nombre del asiento en la base de datos X.500 que contiene la dirección de la interfaz de servicio. Véase 6.7.
type	Object Identifier	El tipo especifica el tipo de servicio. Se supone que sea el identificador de tipo de servicio. En el intermediario X.500, se utiliza para obtener la clase de objeto auxiliar para el nuevo asiento.
properties	Any (Optional)	Los atributos reales X.500 suministrados por el exportador para las propiedades de servicio son controlados por el identificador de tipo de propiedades de servicio.

El T-DUA hace corresponder la operación exportar con una operación X.500 añadir asiento. El nombre del nuevo asiento es seleccionado por el T-DUA de modo que sea único entre todas las ofertas de servicio. Se puede utilizar cualquier esquema; un método simple sería un contador por incrementos. La clase de objeto estructural del nuevo asiento es oferta de servicio La clase de objeto auxiliar es el identificador de tipo de propiedades de servicio (OID) asociado con el identificador de tipo de servicio especificado por el exportador. El T-DUA utiliza su depositario de tipos para obtener el identificador de tipo de interfaz y el identificador de tipo de propiedades de servicio asociados con el tipo de servicio.

Se debe incluir información de control de acceso en el nuevo asiento para impedir que otros usuarios X.500 accedan a la información del intermediario o la modifiquen. Se añade información de control de acceso adicional para que el DSA X.500 pueda controlar el acceso de los clientes a la información del intermediario. Esto se puede ampliar para permitir solamente a determinados clientes el privilegio de exportar servicios.

El parámetro referencia da el nombre de la interfaz que proporciona el servicio. Este asiento es leído para obtener el tipo asociado con la interfaz. Si se puede determinar que este tipo no es un subtipo del tipo de interfaz asociado con el tipo de servicio especificado, se puede devolver una excepción "no hay concordancia de tipo de interfaz". La excepción "referencia de interfaz inválida" se puede aplicar si el parámetro contiene "nil".

Si el intermediario no conoce el identificador de tipo de servicio o la clase de objeto auxiliar, el DSA devuelve un error de actualización del tipo "violación de clase de objeto". El T-DUA devuelve una excepción del tipo "servicio desconocido". El identificador de tipo de servicio es almacenado en el asiento de oferta de servicio.

Los valores de propiedad de servicio que pueden/deben estar presentes en el nuevo asiento son controlados por la clase de objeto auxiliar (identificador de tipo de propiedades de servicio) del asiento. Si el exportador no suministra los atributos obligatorios necesarios para el identificador de tipo de propiedades de servicio especificado, el DSA devuelve un error de actualización del tipo "violación de clase de objeto". El T-DUA devuelve una excepción "falta propiedad obligatoria". De manera similar, si el usuario suministra propiedades de servicio que no están enumeradas en la definición del servicio, se devuelve un error de actualización del tipo "violación de clase de objeto". (La especificación actual en la Rec. UIT-T X.950 | ISO/CEI 13235-1 permite a un exportador incluir una propiedad facultativa ocasional que no está incluida en el tipo de propiedades de servicio. Sin embargo, esto no está permitido en la Recomendación X.500. El exportador debe especificar primero un subtipo que incluye la propiedad adicional antes de la exportación.) El esquema de comprobación en la Recomendación X.500 no verifica que las propiedades de servicio suministradas por el exportador son válidas (por ejemplo, que el tipo de impresora es uno de los tres valores válidos). Esta comprobación tiene que ser efectuada por el T-DUA y se produce la excepción "desigualdad del tipo de propiedad" si el valor de propiedad no es del tipo requerido. Si se produce un error de atributo del tipo atributo indefinido, el T-DUA puede hacer corresponder esto directamente con una excepción "desigualdad del tipo de propiedad".

El T-DUA comprueba también la presencia de propiedades dinámicas en la oferta. Para cada propiedad de servicio dinámica identificada (véase la cláusula 9), el T-DUA sustituye su valor "ficticio" por un valor y crea un valor de propiedad dinámica asociado en el atributo propiedades dinámicas del asiento. El T-DUA fija también el atributo "tiene propiedades dinámicas" del asiento de la oferta a "verdadero", si hay valores dinámicos en la oferta. Si una propiedad con un valor dinámico no es modificable de acuerdo con la información de control de acceso sobre la propiedad, entonces se devuelve una excepción "propiedad dinámica de lectura solamente".

Si un intermediario no admite propiedades dinámicas (es decir, el atributo de intermediario "admite propiedades dinámicas" está puesto a "falso") y hay propiedades dinámicas en la oferta, se produce la excepción "desigualdad de tipo de propiedad" y no se acepta la oferta de servicio.

Si todas las propiedades del servicio no son modificables, el atributo "tiene propiedades modificables" del asiento de oferta se pone a "falso".

Si dos o más propiedades con el mismo nombre de propiedades figuran en la lista de propiedades, se produce la excepción "nombre de propiedad duplicado".

Si la exportación es satisfactoria, el T-DUA devuelve éxito con la siguiente información:

Parámetro	Sintaxis de atributos X.500	Comentario
offerId	DirectoryString	Identificador de oferta de servicio

donde el identificador de oferta (offerId) es la representación en cadena de un nombre de directorio parcial, que da el nombre de trayecto desde el asiento del intermediario a la nueva oferta de servicio. El RDN (sOfferId) del nuevo asiento de oferta de servicio forma el RDN final en la secuencia de nombres.

7.3.2 Retirar

El T-DUA recibe una petición de suprimir una oferta de servicio. La operación retirar corresponde a una operación X.500 suprimir asiento, para suprimir el asiento de oferta de servicio. Los datos de la operación retirar son:

Parámetro	Sintaxis de atributos X.500	Comentario
id	DirectoryString	Un identificador de oferta de servicio

El asiento es identificado por el identificador, que es el DN parcial que identifica inequívocamente a la oferta de servicio. Esta operación suprime simplemente el asiento de oferta de servicio.

Si el cliente suministra un identificador de oferta desconocido, la operación suprimir asiento devuelve un error de nombre de "no hay tal objeto". El T-DUA utiliza entonces el DN parcial para tratar de leer un asiento de oferta por procuración. Si la lectura tiene éxito, se produce la excepción "ningún identificador de oferta de servicio". Si la lectura no tiene éxito, se produce la excepción "identificador de oferta desconocido".

El control de acceso es aplicado por el DSA sobre la base de la información de control de acceso almacenada cuando la oferta de servicio fue exportada y del identificador de cliente de la operación retirar. Si el cliente no tiene suficientes derechos de acceso, se devuelve un error de seguridad de derechos de acceso insuficientes al T-DUA, que corresponde a la excepción específica del sistema. Si la realización requiere que el administrador del intermediario purgue las ofertas "antiguas", el administrador debe tener también permiso para suprimir ofertas.

7.3.3 Modificar

El intermediario recibe una petición de modificar los valores de propiedades de una oferta de servicio. La operación modificar corresponde a una operación X.500 leer y a una operación X.500 modificar asiento, para cambiar el asiento de oferta de servicio. Los datos de la operación modificar son:

Parámetro	Sintaxis de atributos X.500	Comentario
id	DirectoryString	Un identificador de oferta de servicio
del_list	set of OIDs (optional)	Lista de propiedades de servicio que han de ser suprimidas de la oferta
modify_list	Any (optional)	Lista de propiedades de servicio cuyos valores han de ser añadidos o sustituidos

El T-DUA lee primero el asiento de oferta de servicio identificado por el identificador de oferta de servicio. Si este asiento no existe, el DSA devuelve un error de nombre de "no hay tal objeto", que corresponde a una excepción del intermediario "identificador de oferta desconocido". Si el cliente no tiene suficiente permiso para leer el asiento, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes. Esta operación X.500 leer se realiza de

modo que el T-DUA pueda construir una lista de los atributos de propiedad realmente presentes en el asiento de oferta de servicio. El T-DUA modifica el asiento en consecuencia; se suprimen las propiedades de la lista de supresión, se modifican los valores de propiedades existentes y se añaden los nuevos valores de propiedades.

Se construye entonces una operación X.500 modificar asiento, que primero suprime todas las propiedades de servicio existentes y después añade las propiedades de la oferta modificada. Los parámetros de esta nueva oferta de servicio tienen el mismo nombre y función que la operación exportar. Si un nombre de la lista de supresión no existe en la oferta original, se devuelve la excepción "nombre de propiedad desconocido".

El asiento construido recientemente puede violar el esquema (representado por la clase de objeto auxiliar – identificador de tipo de propiedades de servicio). En este caso, cuando se ejecuta la operación X.500 modificar asiento, el DSA devuelve un error de actualización de violación de clase de objeto, y no se modifica la oferta de servicio existente. El T-DUA hace corresponder el error de actualización con diversas excepciones. Se devuelve una excepción "propiedad obligatoria", si una propiedad obligatoria es suprimida pero no se añade de nuevo. El DSA aplica el control de acceso si la operación trata de cambiar el valor de una propiedad no modificable o suprimir una propiedad no modificable, y se devuelve una excepción "lectura solamente"; se devuelve "desigualdad de tipo de propiedad" si los valores no concuerdan con los tipos especificados.

Si la modificación cambia una propiedad no dinámica en una propiedad dinámica, el T-DUA debe añadir el valor de propiedad dinámica apropiado al atributo propiedades dinámicas del asiento. Si la modificación cambia una propiedad dinámica en una propiedad no dinámica, el T-DUA debe suprimir el valor "ficticio" y el valor de propiedad dinámica asociados en el atributo propiedades dinámicas y añadir el nuevo valor estático. Hay que reiniciar también en consecuencia la bandera "tiene propiedades dinámicas".

Los resultados de añadir las nuevas propiedades equivalen a exportar la oferta de servicio, y el análisis de los errores que se pueden producir según 7.3.1 se aplica también en este caso.

La invocación de la operación modificar asiento hace que el T-DUA devuelva una excepción "no realizada" si el atributo de intermediario "admite propiedades modificables" está puesto a "falso".

7.3.4 Describir

El T-DUA recibe una petición de describir una oferta de servicio. La operación describir corresponde a una operación X.500 leer, para leer el asiento de oferta de servicio. Los datos de la operación describir son:

Parámetro	Sintaxis de atributos X.500	Comentario
id	DirectoryString	Un identificador de oferta de servicio

Para la operación leer, el asiento es identificado por el identificador (el DN parcial de un asiento de oferta de servicios) con la información de asiento puesta a "todos los atributos de usuario". El T-DUA debe examinar el atributo de propiedades dinámicas del asiento devuelto. Para cada valor de propiedad dinámica fijado en el atributo propiedades dinámicas, se debe suprimir el valor "ficticio" asociado y sustituirlo por el valor devuelto en la operación evaluar propiedades dinámicas invocada en la interfaz de evaluación de propiedades dinámicas asociada, para extraer el valor actual de esa propiedad (véase 7.9).

El T-DUA filtra los atributos devueltos y transfiere a su cliente la siguiente información:

Parámetro	Sintaxis de atributos X.500	Comentario
reference	Name	Nombre de la interfaz que proporciona el servicio
type	Object Identifier	El identificador de tipo de servicio de la oferta
properties	Set of attributes	Cada tipo de atributo corresponde a un nombre de propiedad y el correspondiente valor del atributo corresponde a un valor de propiedad

Si el cliente suministra un identificador de oferta de servicio desconocido, se devuelve un error de nombre de "no hay tal objeto". El T-DUA devuelve una excepción "identificador de oferta desconocido".

El DSA aplica el control de acceso basado en la información de control de acceso almacenada cuando se exportó la oferta de servicio y en el identificador de cliente de la operación describir. Si el cliente no tiene suficientes derechos de acceso, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes que corresponde al error de sistema apropiado.

7.3.5 Retirar con restricción

El T-DUA recibe una petición de suprimir un conjunto de ofertas de servicio dentro de un intermediario. Este conjunto se identifica de la misma manera que una operación indagar (véase 7.4.1) identifica un conjunto de ofertas que se han de devolver a un importador.

Los datos de la operación retirar con restricción son:

Parámetro	Sintaxis de X.500	Comentario
type	Object Identifier	Identifica al identificador de tipo de servicio que se ha de usar para seleccionar las ofertas que han de ser suprimidas
constr	Filter	La restricción utilizada para seleccionar las ofertas que han de ser suprimidas

El primer paso para procesar esta operación es construir un filtro X.500 a partir de la restricción. Esto se complica debido a las propiedades dinámicas en las ofertas de servicio. Cada prueba de propiedad en la restricción es ampliada para ver si el valor de propiedad dinámica asociado está presente. Por ejemplo, la prueba "dotsPerInch = 600" se extiende a "dotsPerInch = 600 OR dotsPerInchDynamicPropValue present". El filtro debe incluir la concordancia del "tipo" del servicio.

Una vez construido el filtro X.500, el T-DUA invoca una operación X.500 buscar, para comparar asientos de ofertas de servicio. La información de asiento para la operación X.500 buscar es seleccionada en un identificador de oferta de servicio (sOfferId,) que es el RDN de una oferta que concuerda.

La Recomendación X.500 no informa error si una propiedad de servicio en la restricción especificada por el cliente no es reconocida por el DSA. De manera similar, no se informa error si un nombre de propiedad de servicio no es reconocido por el DSA.

La operación X.500 suprimir asiento se utiliza para suprimir cada una de las ofertas de servicio comparadas por el DSA. La operación suprimir asiento suprime simplemente el asiento de oferta de servicio. El asiento es identificado por el identificador de oferta de servicio que es devuelto por el DSA para las ofertas comparadas.

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando la oferta de servicio fue exportada y del identificador de cliente de la operación retirar. Si el cliente no tiene suficientes derechos de acceso, se devuelve un error de seguridad de derechos de acceso insuficientes al T-DUA, que corresponde a la excepción específica del sistema. Si la realización requiere que el administrador del intermediario purgue las ofertas "antiguas", el administrador debe tener también permiso para suprimir ofertas.

Se devuelve una excepción "ninguna oferta concuerda" si la operación X.500 buscar no da ofertas que concuerdan.

7.3.6 Resolver

El T-DUA recibe una petición de resolver un nombre relativo de contexto de otro intermediario. En particular, esta operación se utiliza cuando se exporta a un intermediario que es conocido por su nombre (una secuencia de nombres de enlaces que identifican un trayecto de intermediarios vinculados en un gráfico de intermediación), en vez de ser conocido por una referencia de interfaz. El cliente proporciona el nombre de una secuencia de componentes de nombre. Cada componente se utiliza para comparar un nombre de enlace con un gráfico de intermediación. Los datos de la operación describir son:

Parámetro	Sintaxis de atributos X.500	Comentario
name	A sequence of Names	Un nombre de intermediario formado por una secuencia de nombres, donde cada nombre es un nombre de enlace

El T-DUA extrae y suprime el primer componente de la secuencia (un nombre de enlace) y utiliza este componente como un filtro para "nombre de enlace" en una operación de búsqueda de un nivel en sus asientos de enlace. La información del asiento para la operación X.500 buscar se selecciona en "identificador de interfaz de intermediario deseado".

Si existe otro nombre en la secuencia, el T-DUA utiliza una operación X.500 leer para hacer corresponder el "identificador de interfaz de intermediario deseado" hallado con una dirección del intermediario deseado. El T-DUA se conecta con el intermediario deseado vinculado e invoca una operación resolver con los nombre residuales como su parámetro. Este proceso continúa hasta que la secuencia está vacía, en cuyo punto, el T-DUA (que ejecuta la última búsqueda) devuelve el "identificador de interfaz de intermediario deseado" final a su cliente, que a su vez devuelve la dirección a su cliente, y así sucesivamente. A la larga, la dirección de interfaz de intermediario final correspondiente al nombre de intermediario final es devuelta al invocador original de esta operación.

Se devuelve una excepción "nombre de intermediario desconocido" si cualquiera de las operaciones X.500 buscar no tiene éxito a lo largo del trayecto.

Si el cliente no tiene permiso suficiente para buscar los asientos de enlaces, se devuelve un error de seguridad de derechos de acceso insuficientes al T-DUA que lo hace corresponder con una excepción específica del sistema.

7.4 Operaciones consultar

Las operaciones consultar proporcionan la funcionalidad de importación para la función intermediación. Indagar es la única operación. Se utiliza para importar asientos de ofertas de servicio almacenadas en un intermediario o en un grupo de intermediarios federados.

7.4.1 Operación indagar

El intermediario recibe una petición de buscar ofertas de servicio que concuerden con el tipo de servicio y restricción especificadas por el cliente, con un alcance de búsqueda previsto identificado por la política de importador. Facultativamente, el conjunto de oferta comparadas son filtradas por los criterios de preferencia del cliente. La operación indagar puede incluir también los nombres de aquellas propiedades cuyos valores son de interés para el importador. Los datos de la operación indagar son:

Parámetro	Sintaxis de atributos X.500	Comentario
type	Object Identifier	El tipo especifica el servicio que ha de concordar
constr	Filter	Representa la restricción utilizada para los criterios de concordancia de la oferta, véase más adelante
pref	No se utiliza	Criterios de preferencia, utilizados por el T-DUA solamente
policies	(Optional)	Expectativas del importador, véase más adelante
desired_props	(Optional)	Propiedades del servicio de interés, véase más adelante
how_many	No se utiliza	Control de iterador utilizado por el T-DUA solamente

Debido a la complejidad de una importación, la descripción se divide en dos componentes: búsqueda del espacio de ofertas de servicio del intermediario local y búsqueda de los espacios de ofertas de intermediarios federados. A elección del T-DUA, estas dos búsquedas se pueden efectuar secuencialmente o en paralelo.

7.4.2 Políticas

El primer paso para procesar una operación indagar es procesar las políticas. Las políticas se representan como pares de <nombre, valor>. Un importador puede especificar ninguna o más políticas de importador que indiquen las expectativas del importador para la búsqueda. Cada importador tiene sus propias políticas que orientan su comportamiento al respecto. Las políticas de intermediario interactúan con las políticas de importador. Si no se especifica una política de importador, el intermediario utiliza la política de intermediario por defecto. Si una política de importador rebasa los valores de política máximos (o límite) fijados por el intermediario, éste sustituye las expectativas del importador con el valor de política máximo/límite del intermediario.

Es responsabilidad del T-DUA verificar y fijar los límites superiores reales para las políticas de cardinalidad y la política de cómputo de saltos. Es también responsabilidad del T-DUA colocar un nombre de política en el parámetro devuelto "límites aplicados" si una búsqueda es terminada prematuramente por un límite superior.

Un intermediario sustenta tres políticas normalizadas de capacidades. Si un intermediario no sustenta una capacidad, el importador no puede sustituirla; la no utilización de una capacidad solicitada por un importador se informa incluyendo el nombre de política de capacidades en "límites aplicados". Sin embargo, si un intermediario sustenta una capacidad y un importador no desea utilizarla, el intermediario debe respetar el deseo del importador. La capacidad del intermediario se aplica si un importador no especifica una capacidad prevista. Hay tres opciones de capacidades normalizadas para un importador:

- utilizar ofertas por procuración – si está puesto a "falso", no se buscan los asientos de ofertas por procuración para comparación;
- utilizar propiedades modificables – si está puesto a "falso", el T-DUA añade el término "tiene propiedades modificables = falso" en el filtro de concordancia;
- utilizar propiedades dinámicas – si está puesto a "falso", el T-DUA añade el término "tiene propiedades dinámicas = falso" en el filtro de concordancia.

El siguiente paso es procesar la política de comportamiento de enlaces para determinar si puede haber algún interfuncionamiento. Esto se describe en 7.4.4.

Si un importador indica una política de comienzo por intermediario, se compara el primer componente en el nombre de intermediario con el nombre mantenido en cada asiento de enlace. Si no se encuentra concordancia, se devuelve la excepción "valor de política inválido". En los demás casos, el T-DUA invoca la operación indagar en el enlace denominado y transfiere la política de "comienzo por intermediario" suprimiendo el primer componente (véase también 7.3.6 para la operación resolver).

7.4.3 Búsqueda local

La búsqueda local comienza con la construcción de un filtro X.500 para comparar el identificador de tipo de servicio. Además de las ofertas de servicio que concuerdan exactamente con el identificador de tipo de servicio, el filtro puede comparar también cualquier asiento con un identificador de tipo de servicio que es un refinamiento (subclase) del tipo especificado (identificador de tipo de servicio). Sin embargo, solamente se efectúa una concordancia de tipo de servicio exacta si el importador ha fijado la política de importador "concordancia de tipo exacta" a "verdadero" en la operación indagar. Si el intermediario admite propiedades dinámicas y el importador desea utilizar ofertas con propiedades dinámicas, se selecciona el atributo "tiene propiedades dinámicas" para la selección de información de asiento en la operación X.500 buscar. Se ejecuta una búsqueda de un nivel en el asiento de ofertas de servicio para ver si hay alguna oferta de servicio con el tipo de servicio requerido.

Se construye un segundo filtro a partir de la construcción especificada para la concordancia. Esto puede ser complicado por la sustentación y presencia de propiedades dinámicas en las ofertas de servicio (conocimiento adquirido en la primera búsqueda). Si el intermediario admite propiedades dinámicas y el importador desea utilizar ofertas con propiedades dinámicas, cada prueba de propiedad en la construcción se amplía para saber si el "valor de propiedad dinámica" asociado está presente. Por ejemplo, la prueba "dotsPerInch = 600" se extiende a "dotsPerInch = 600 OR dotsPerInchDynamicPropValue present".

El segundo filtro se pone a "Y" con el primer filtro. El T-DUA invoca una operación X.500 buscar para comparar los asientos de ofertas de servicio. Todos los atributos de usuario son seleccionados en la selección de información de asiento X.500; es responsabilidad del T-DUA filtrar los atributos devueltos. Es posible que un usuario desee tener la descripción de valores para "ninguna", "algunas" o "todas" las propiedades del servicio.

La Recomendación X.500 no informa error si una propiedad de servicio en la construcción especificada por el cliente no es reconocida por el DSA. De manera similar, no se informa error si un nombre de propiedad de servicio no es reconocido por el DSA. Sin embargo, se devuelve un error de atributo de "no hay tal atributo o valor" si no se devuelven atributos.

Si un DSA compara muchas ofertas de servicio contra el filtro, el DSA sólo tiene que devolver algunas de las concordancias. En este caso, se fija el calificador de salida parcial "límite de tamaño rebasado". El T-DUA puede tener que recomenzar la búsqueda, utilizando la opción de resultados de la operación X.500 buscar.

Los asientos devueltos por la operación X.500 buscar pueden contener propiedades dinámicas. El T-DUA debe examinar cada asiento con respecto al atributo "propiedades dinámicas". Si se encuentra, el atributo ficticio asociado para cada valor de propiedad dinámica en el atributo propiedades dinámicas se suprime del resultado de la búsqueda y se invoca la interfaz asociada para evaluación de la propiedad, con el fin de extraer el valor actual de esa propiedad dinámica (para la evaluación de las propiedades dinámicas, véase 7.9). La restricción de indagar debe ser evaluada por el T-DUA para determinar si las ofertas de servicio realmente satisfacen la restricción.

7.4.4 Búsqueda de intermediario federados

Un T-DUA utiliza un contador de saltos para controlar la profundidad de los enlaces atravesados por una importación. Un cómputo de enlaces de cero termina la propagación de una importación a través de intermediarios vinculados. Cada vez que se atraviesa un enlace para transferir una operación indagar, se sustrae un uno del contador de enlaces.

Hay dos políticas de seguimiento de enlace asociadas con cada enlace, regla de seguimiento de transferencia por defecto (`def_pass_on_follow_rule`) y regla limitadora de seguimiento (`limiting_follow_rule`), que se fijan en el momento de creación del enlace. El propio intermediario tiene tres políticas conexas de enlace:

- política de seguimiento de enlace máximo – limita el valor de la regla limitadora de seguimiento permitido cuando un enlace se establece por primera vez;
- política de seguimiento máximo – limita el comportamiento de seguimiento del enlace en cualquier enlace del intermediario para una operación indagar dada;
- política de seguimiento por defecto – proporciona la regla de seguimiento de enlace por defecto si el importador no especifica ninguna en su operación indagar.

Por consiguiente, la política de seguimiento para un determinado enlace es:

- `min (trader.max_follow_policy, link.limiting_follow_rule, query.link_follow_rule)`; o
- `min (trader.max_follow_policy, link.limiting_follow_rule, trader.def_follow_policy)`.

Si ha de efectuar el seguimiento de algunos enlaces adecuados, el T-DUA utiliza la operación X.500 leer para hacer corresponder los nombres de identificador de interfaz de intermediario deseado (de los asientos de enlace de intermediarios) con la dirección de cada intermediario deseado. El T-DUA se conecta a los intermediarios seleccionados y les transfiere la operación indagar con la regla de seguimiento de enlace especificada como:

- `min (trader.max_follow_policy, link.limiting_follow_rule, query.link_follow_rule)`; o
- `min (trader.max_follow_policy, link.def_pass_on_follow_rule)`.

Además, si el intermediario de origen que inicia una operación indagar a intermediarios federados desea incluir un identificador para la operación indagar en el parámetro de política, puede hacerlo así. El intermediario no está obligado a generar este identificador de operación, pero está obligado a transferir esta política al enlace.

7.4.5 Búsqueda de ofertas por procuración

Si la política "utilizar ofertas por procuración" del importador no está puesta a "falso" y el atributo "admite ofertas por procuración" del intermediario está puesto a "verdadero", se repiten las búsquedas X.500 indicadas en 7.4.3 en los asientos de ofertas por procuración. Para cada oferta por procuración comparada, el T-DUA aplica una operación indagar modificada utilizando la información almacenada en la oferta por procuración que concuerda. El T-DUA utiliza entonces una operación X.500 leer para hacer corresponder el nombre de identificador de interfaz de consulta por procuración (del asiento de oferta por procuración) con una "dirección" que el T-DUA utiliza para invocar la operación indagar modificada.

7.4.6 Oferta de servicio devuelta

Cuando el intermediario encuentra un conjunto de ofertas de servicio, el parámetro de preferencia y las diversas política de cardinalidad se utilizan para ordenar y limitar las ofertas devueltas al importador. Ésta es una función del T-DUA y no se considera más detalladamente en esta especificación. La realización de un iterador de ofertas es también responsabilidad del T-DUA.

Se devuelve la excepción "tipo desconocido" (nombre de tipo de servicio no conocido) o "ningún tipo concuerda" (no hay ofertas con el tipo de servicio requerido) si cualquiera de estos dos es verdadero en el espacio de búsqueda total.

7.5 Operaciones de enlace

7.5.1 Añadir enlace

El intermediario recibe una operación añadir enlace, que corresponde con una operación X.500 añadir asiento, para añadir un nuevo asiento de enlace de intermediario en el directorio X.500. Los datos de la operación añadir enlace son:

Parámetro	Sintaxis de atributos X.500	Comentario
name	DirectoryString	Nombre del nuevo enlace
target	Name	Nombre de otro intermediario
def_pass_on_follow_rule	FollowOption	No debe rebasar la regla limitadora de seguimiento
limiting_follow_rule	FollowOption	No debe rebasar la política de seguimiento de enlace máximo del intermediario

El T-DUA comprueba primero los valores admisibles para los parámetros. Se produce la excepción "seguimiento por defecto demasiado permisivo" si las reglas de seguimiento de transferencia por defecto exceden de la regla limitadora de seguimiento. Se produce la excepción "seguimiento límite demasiado permisivo" si la regla limitadora de seguimiento excede de la política de seguimiento de enlace máximo del intermediario. El T-DUA debe leer primero el asiento del intermediario para determinar su política de seguimiento de enlace máximo. El T-DUA se asegura también de que el nuevo nombre de enlace es único dentro del conjunto de enlaces de intermediario buscando un caso existente del nombre. Si se encuentra un nombre que concuerda, se produce la excepción "nombre de enlace duplicado".

El T-DUA hace corresponder la operación añadir enlace con una operación X.500 añadir asiento y selecciona el identificador del nuevo asiento, de modo que sea único entre todos los enlaces de intermediario. Se puede utilizar cualquier esquema; un método sencillo sería un contador por incrementos. La clase de objeto estructural del nuevo asiento es enlace de intermediario. La clase de objeto auxiliar enlace de intermediario puede especificar también propiedades de enlace adicionales que pueden ser incluidas en la operación añadir asiento.

Se debe incluir información de control de acceso en el nuevo asiento para impedir que otros usuarios de la Recomendación X.500 o del intermediario accedan a la información de enlace o la modifiquen.

Si la operación añadir enlace tiene éxito, no se producen excepciones.

7.5.2 Suprimir enlace

El T-DUA recibe una petición de suprimir un enlace de intermediario. Los datos de la operación suprimir enlace son:

Parámetro	Sintaxis de atributos X.500	Comentario
name	DirectoryString	Identifica el enlace que ha de ser suprimido por su nombre de enlace

El T-DUA debe utilizar primero la operación X.500 buscar, para encontrar en los asientos de enlace un asiento con ese nombre de enlace. Si la búsqueda fracasa, se produce la excepción "nombre de enlace desconocido".

El T-DUA utiliza entonces una operación X.500 suprimir asiento, para suprimir el asiento de enlace de intermediario utilizando el identificador de enlace del asiento que concuerda. Esta operación suprime sencillamente el asiento de enlace de intermediario.

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando se creó el enlace de intermediario y del identificador de cliente de la operación suprimir. Si el cliente no tiene derechos de acceso suficientes, se devuelve un error de seguridad de derechos de acceso insuficientes al T-DUA, que lo hace corresponder con una excepción específica del sistema.

7.5.3 Modificar enlace

El intermediario recibe una petición de modificar un enlace de intermediario. La operación modificar corresponde a una operación X.500 buscar, para hallar el enlace que utiliza el nombre de enlace, y a una operación modificar asiento, para cambiar el asiento de enlace de intermediario. Los datos de la operación modificar enlace son:

Parámetro	Sintaxis de atributos X.500	Comentario
name	DirectoryString	Identifica al enlace que ha de ser modificado
def_pass_on_follow_rule	FollowOption	No debe rebasar la regla limitadora de seguimiento
limiting_follow_rule	FollowOption	No debe rebasar la política limitadora de seguimiento del intermediario

El T-DUA comprueba primero los valores admisibles de los parámetros. Se produce la excepción "seguimiento por defecto demasiado permisivo" si la regla de seguimiento de transferencia por defecto excede de la regla limitadora de seguimiento. Se produce la excepción "seguimiento límite demasiado permisivo" si la regla limitadora de seguimiento excede de la política de seguimiento de enlace máximo del intermediario. El T-DUA busca el asiento de enlace de intermediario identificado por el nombre de enlace. Se seleccionan todos los atributos de usuario en la selección de información de asiento X.500. Si este asiento no existe, el DSA devuelve un error de nombre de "no hay tal objeto", que corresponde a la excepción "nombre de enlace desconocido".

Se construye después una operación X.500 modificar asiento, que suprime primero todo el comportamiento de seguimiento de enlaces existente (y las propiedades, si las hubiere) y añade el nuevo comportamiento de seguimiento (y las propiedades, si las hubiere) de la operación modificar enlace. Si el cliente no tiene permiso suficiente para suprimir atributos de ese asiento, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes.

La clase de objeto auxiliar enlace de intermediario puede especificar también propiedades de enlace que se pueden incluir en la operación modificar asiento. El asiento recientemente modificado puede violar el esquema. En este caso, se añade el asiento de enlace recientemente modificado y el DSA devuelve un error de actualización de violación de clase de objeto. No se modifica el enlace de intermediario existente. La comprobación del esquema X.500 no puede verificar que los valores son los especificados por el esquema de enlace. Se produce un error de atributo de tipo de atributo indefinido si el DSA no conoce uno de los atributos suministrados por el cliente.

7.5.4 Describir enlace

El intermediario recibe una petición de devolver detalles de un determinado enlace. Esta operación corresponde a una operación X.500 buscar. Los datos son:

Parámetro	Sintaxis de atributos X.500	Comentario
Name	DirectoryString	Identifica el enlace que ha de ser extraído

El T-DUA busca el asiento de enlace de intermediario identificado por el nombre de enlace con la selección de información de asiento puesta a "todos los atributos de usuario". Si este asiento no existe, el DSA devolverá un error de nombre de "no hay tal objeto", que corresponde a una excepción "nombre de enlace desconocido". Si el cliente no tiene permiso suficiente para buscar el asiento, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes.

Una vez que el DSA X.500 ha devuelto los datos, el T-DUA devuelve al cliente los datos, menos el identificador de enlace.

7.5.5 Enumerar enlaces

El intermediario recibe una petición de devolver los nombres de todos los enlaces de intermediario conocidos. Esta operación corresponde a la operación X.500 buscar. Los datos son:

Parámetro	Sintaxis de atributos X.500	Comentario
(ninguno)		

El T-DUA ejecuta una operación X.500 buscar para devolver el atributo nombre de enlace de todos los asientos de enlace de intermediarios. Es una búsqueda de un nivel en los asientos de enlace. Los nombres de enlace (si los hubiere) se devuelven al usuario:

Parámetro	Sintaxis de atributos X.500	Comentario
LinkNameSeq	Set of DirectoryString	

donde cada cadena de directorio (un nombre de enlace) es única para cada asiento de enlace.

7.6 Operaciones de ofertas por procuración

Las operaciones de ofertas por procuración forman un conjunto de operaciones que permiten a un cliente exportar, retirar y describir ofertas por procuración en el intermediario.

7.6.1 Exportar por procuración

El intermediario recibe una operación exportar por procuración, que corresponde a una operación X.500 añadir asiento, para añadir un nuevo asiento de oferta por procuración en el directorio X.500. Los datos de esta operación son:

Parámetro de intermediario	Sintaxis de atributos X.500	Comentario
type	Object Identifier	Se supone que el tipo de nombre sea el identificador de tipo de servicio. Se usa para obtener la clase de objeto auxiliar para el nuevo asiento.
target	Name	El nombre del asiento en la base de datos X.500 que contiene la dirección del identificador de interfaz de consulta por procuración.
properties	Any (optional)	Los atributos reales X.500 suministrados por el exportador para las propiedades de servicio son controlados por el identificador de propiedades de servicio.
if_match_all	Boolean	
recipe	String	
policies_to_pass_on	Set of Attributes (optional)	

El T-DUA hace corresponder la operación exportar por procuración con una operación X.500 añadir asiento. El T-DUA selecciona el nombre del nuevo asiento de modo que sea único entre todas las ofertas por procuración. Se puede utilizar cualquier esquema; un método sencillo sería un contador por incrementos. La clase de objeto estructural del nuevo asiento es oferta por procuración. La clase de objeto auxiliar es el identificador de tipo de propiedades de servicio (OID) asociado con el parámetro identificador de tipo de servicio de la oferta por procuración. El T-DUA utiliza su depositario de tipos para obtener el identificador de tipo de interfaz y el identificador de tipo de propiedades de servicio asociado con este tipo de servicio.

Se debe incluir información de control de acceso en el nuevo asiento para impedir que otros usuarios de la Recomendación X.500 accedan a la información de intermediario o la modifiquen. Se añade información de control de acceso adicional para que el DSA X.500 pueda controlar el acceso de los clientes a la información de intermediario. Esto se puede ampliar para que sólo determinados clientes tengan el privilegio de exportar ofertas por procuración.

ISO/CEI 13235-3 : 1998 (S)

Si el intermediario no conoce el identificador de tipo de servicio o clase de objeto auxiliar, el DSA devuelve un error de actualización del tipo violación de clase de objeto de tipo. El T-DUA devuelve una excepción "tipo de servicio desconocido". El identificador de tipo de servicio se almacena en el asiento oferta por procuración.

Los valores de propiedades de servicio que pueden/deben estar presentes en el nuevo asiento son controlados por la clase de objeto auxiliar (identificador de tipo de propiedades de servicio) del asiento. El procesamiento de estos valores de propiedad son iguales que para la operación exportar oferta de servicio (véase 7.3.1).

El parámetro "si todo concuerda" especifica si esta oferta sólo requiere concordancia del tipo de servicio.

El parámetro objetivo da el nombre de la interfaz que se utiliza para obtener la interfaz de servicio en el tiempo de ejecución. Si se especifica "nil", se produce la excepción "referencia de interfaz inválida".

El parámetro fórmula proporciona la modificación que se puede aplicar a la construcción de una operación indagar que compara esta oferta por procuración en cuanto al tipo de servicio y valores de propiedades cuando "si todo concuerda" no es "verdadero".

Las políticas de transferir (_to_pass_on) consisten en pares de <nombre, valor> que se han de añadir al parámetro de políticas de la operación indagar entrante para formar el parámetro de política modificada de la operación indagar modificada.

Si la exportación de la oferta por procuración tiene éxito, el T-DUA devuelve éxito con la siguiente información:

Parámetro	Sintaxis de atributos X.500	Comentario
OfferId	DirectoryString	Un identificador de oferta por procuración

donde el identificador de oferta es la representación en cadena del DN parcial del nuevo asiento de oferta por procuración.

7.6.2 Retirar por procuración

El T-DUA recibe una petición de suprimir una oferta por procuración. La operación retirar por procuración corresponde a la operación X.500 suprimir asiento, para suprimir el asiento de oferta por procuración. Los datos de la operación retirar por procuración son:

Parámetro	Sintaxis de atributos X.500	Comentario
id	DirectoryString	Un identificador de oferta por procuración

Esta operación suprime sencillamente el asiento de oferta por procuración. El asiento es identificado por el "id" que identifica inequívocamente un asiento de oferta por procuración.

Si el cliente suministra un identificador de oferta por procuración desconocido, la operación suprimir asiento devuelve un error de nombre de "no hay tal objeto". El T-DUA utiliza entonces el identificador de oferta por procuración para leer un asiento de oferta de servicio. Si la lectura tiene éxito, se produce la excepción "ningún identificador de oferta por procuración". Si la lectura fracasa, se produce la excepción "identificador de oferta desconocido".

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando se exportó la oferta por procuración y del identificador de cliente de la operación retirar por procuración. Si el cliente no tiene derechos de acceso suficientes, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes, que corresponde a la excepción específica del sistema. Si la realización requiere que el administrador del intermediario purgue las ofertas por procuración "antiguas", el administrador debe también tener permiso para suprimir ofertas por procuración.

7.6.3 Describir por procuración

El T-DUA recibe una petición de describir una oferta por procuración. La operación describir corresponde a la operación X.500 leer, para leer el asiento de oferta por procuración. Los datos de esta operación son:

Parámetro	Sintaxis de atributos X.500	Comentario
id	DirectoryString	Un identificador de oferta por procuración

El asiento es identificado por el "identificador" que identifica inequívocamente el asiento. La información del asiento para la operación de lectura se pone a "todos los atributos de usuario". El T-DUA no tiene que evaluar los valores de propiedades dinámicas en la oferta por procuración.

El T-DUA filtra los atributos devueltos y transfiere a su cliente la siguiente información:

Parámetro	Sintaxis de atributos X.500	Comentario
type	OID	Identificador de tipo de servicio
target	DN	Nombre del asiento de interfaz que proporciona la referencia de la interfaz de servicio
serviceProperties	Set of attributes	Un tipo de atributo corresponde a un nombre de propiedad y un valor de atributo corresponde a un valor de propiedad
if_no_match	boolean	
recipe	string	
policies_to_pass_on	Set of attributes (if any)	

Si el cliente suministra un identificador de oferta por procuración desconocido, la operación leer devuelve un error de nombre de "no hay tal objeto". El T-DUA utiliza entonces el identificador de oferta por procuración para tratar de leer un asiento de oferta de servicio. Si la lectura tiene éxito, se produce la excepción "ningún identificador de oferta por procuración". Si la lectura fracasa, se produce la excepción "identificador de oferta desconocido".

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando se exportó la oferta por procuración y del identificador de cliente de la operación describir por procuración. Si el cliente no tiene suficientes derechos de acceso, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes que corresponde al error de sistema apropiado.

7.7 Operaciones de atributos de intermediarios

Las propiedades y las políticas de intermediario se expresan como atributos de la Recomendación X.500 en el asiento de intermediario. Todos los clientes registrados de un intermediario pueden "obtener" información sobre las características del intermediario utilizando la operación X.500 leer en el asiento del intermediario, para los atributos de características. Además, todos los exportadores registrados pueden tener los derechos de acceso para leer los siguientes atributos: admite propiedades modificables, admite propiedades dinámicas, admite ofertas por procuración e información de depositarios de tipos. Todos los importadores registrados pueden, además, tener los derechos de acceso para leer los siguientes atributos de intermediario: tarjeta de búsqueda por defecto, tarjeta de búsqueda máxima, tarjeta de concordancia por defecto, tarjeta de concordancia máxima, tarjeta de devolución por defecto, tarjeta de devolución máxima, lista máxima, cómputo de salto por defecto, cómputo de saltos máximo, admite propiedades modificables, admite propiedades dinámicas, admite ofertas por procuración, política de seguimiento por defecto, política de seguimiento máximo e interfaz de depositario de tipos.

El administrador del intermediario puede "obtener" y "fijar" todos los atributos de intermediario, donde la operación "obtener" corresponde a la operación X.500 leer y la operación "fijar" corresponde a la operación X.500 modificar asiento.

7.8 Operaciones administrativas

7.8.1 Enumerar ofertas

El intermediario recibe una petición de devolver los identificadores de todas las ofertas de servicio conocidas por el intermediario. Esta operación corresponde a la operación X.500 buscar. Los datos son:

Parámetros	Sintaxis de atributos X.500	Comentario
how_many	No se utiliza	Este parámetro es mantenido por el T-DUA en su realización del iterador de Id

ISO/CEI 13235-3 : 1998 (S)

El T-DUA ejecuta una operación X.500 buscar para devolver los identificadores de todos los asientos de ofertas de servicio. Es una búsqueda de un nivel para cualquier asiento de oferta de servicio. No se solicita información. Estos identificadores (si los hubiere) se devuelven al usuario como:

Parámetro	Sintaxis de atributos X.500	Comentario
ids	Set of DirectoryString	

donde cada identificador (representación en cadena de un DN parcial) identifica inequívocamente a un asiento de oferta de servicio.

El T-DUA es responsable de aplicar el iterador de identificadores.

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando la oferta de servicio se exportó y sólo los administradores pueden ejecutar la operación enumerar ofertas. Si un cliente no tiene derechos de acceso suficientes, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes, que corresponde al error de sistema apropiado.

7.8.2 Enumerar procuraciones

El intermediario recibe una petición de devolver los identificadores de todas las ofertas por procuración conocidas por el intermediario. Esta operación corresponde a la operación X.500 buscar. Los datos son:

Parámetros	Sintaxis de atributos X.500	Comentario
how_many	No se utiliza	Este parámetro es mantenido por el T-DUA en su realización del iterador de identificador

El T-DUA ejecuta una operación X.500 buscar para devolver los identificadores de todos los asientos de ofertas de servicio. Es una búsqueda de un nivel para cualquier asiento de oferta de servicio. No se solicita información. Estos identificadores (si los hubiere) se devuelven al usuario como:

Parámetro	Sintaxis de atributos X.500	Comentario
ids	Set of DirectoryString	

donde cada identificador (representación en cadena de un DN parcial) identifica inequívocamente a un asiento de oferta por procuración.

El T-DUA es responsable de aplicar el iterador de identificadores.

El DSA aplica el control de acceso sobre la base de la información de control de acceso almacenada cuando la oferta por procuración se exportó y sólo los administradores pueden ejecutar la operación enumerar ofertas por procuración. Si un cliente no tiene derechos de acceso suficientes, se devuelve al T-DUA un error de seguridad de derechos de acceso insuficientes, que corresponde al error de sistema apropiado.

7.9 Operaciones de evaluación de propiedades dinámicas

El T-DUA actúa como un cliente para las operaciones de evaluación de propiedades dinámicas con el fin de obtener valores para las propiedades dinámicas. La única operación especificada es evaluar propiedades dinámicas.

7.9.1 Evaluar propiedades dinámicas

Para obtener el valor de una propiedad dinámica en el tiempo de ejecución, se especifica la signatura para la operación "evaluar propiedades dinámicas". Los datos de la operación son:

Parámetro	Sintaxis de atributos X.500	Comentario
name	Object Identifier	Identifica el nombre de propiedad
returned_type	Object identifier	Identifica el tipo de propiedad
extra_info	Any (optional)	Puede incluir información requerida para obtener el valor de propiedades

La operación devuelve "cualquier" valor que deba contener un valor para la propiedad con un tipo de "tipo devuelto". Se produce la excepción "fallo de evaluación de propiedad dinámica" si no se puede determinar el valor de la propiedad.

La forma en que el valor de una propiedad dinámica se obtiene realmente no se especifica en la Rec. UIT-T X.950 | ISO/CEI 13235-1.

8 Depositario de tipos

El concepto del depositario de tipos es central para el intermediario. En esta sección se define un depositario de tipos mínimo, que sólo abarca aquellas partes del depositario de tipos que son necesarias para permitir el funcionamiento correcto del directorio X.500.

NOTA 1 – Es factible realizar depositarios de tipos más complicados, pero el trabajo en este campo se convierte rápidamente en la investigación de depositarios de tipos en vez de la especificación de un intermediario que utiliza el directorio X.500.

NOTA 2 – En el momento de la publicación de la presente especificación se está trabajando sobre la función de depositario de tipos ODP.

8.1 Esquema X.500 y el depositario de tipos mínimo

El depositario de tipos mínimo consiste en el esquema X.500 normal. Las funciones proporcionadas por este componente (y las limitaciones) se describen a continuación.

Cada identificador de tipo de servicio es un identificador de objeto (OID, *object identifier*) e identifica un tipo de servicio. Hay un tipo de interfaz (identificada por un OID) y un tipo de propiedades de servicio (también identificado por un OID) asociados con cada tipo de servicio. El tipo de interfaz identifica las operaciones (y sus parámetros asociados) para el servicio y el tipo de propiedades de servicio identifica las propiedades de servicio asociadas con el servicio.

Cuando el T-DUA recibe un identificador de tipo de servicio de su exportador, envía el OID a su depositario de tipos y recibe de éste el identificador de tipo de interfaz (OID) y el identificador de tipo de propiedades de servicio (OID) asociados. El T-DUA almacena el identificador de tipo de servicio en el asiento de oferta de servicio y utiliza el identificador de tipo de propiedades de servicio como la clase de objeto auxiliar para ese asiento de oferta de servicio.

La clase de objeto auxiliar define los atributos de propiedad que pueden ser incluidos en una oferta de servicio y lo hace mediante dos listas:

- La lista "debe contener", que contiene los atributos de propiedades que deben estar presentes en la oferta de servicio.
- La lista "puede contener", que contiene los atributos de propiedades que pueden estar presentes en la oferta de servicio.

Los atributos de propiedades que no están presentes en ninguna de las dos listas no pueden ser incluidos en las propiedades de servicio de la oferta de servicio. La utilización apropiada de las reglas de control de acceso para una propiedad determina si una propiedad particular es de lectura solamente y no puede ser modificada ni suprimida. El tratamiento de las propiedades dinámicas se examina en la cláusula 9.

Los atributos de propiedades son identificados también por los OID. Las definiciones de atributos pueden controlar:

- Si se pueden almacenar múltiples valores en el atributo.
- La sintaxis (por ejemplo, cadena, entero) de los valores. Se pueden definir también las sintaxis de atributos complejos (por ejemplo, un valor consiste en una cadena seguida por un entero). Alguna sintaxis (por ejemplo, sintaxis de números telefónicos) imponen restricciones en cuanto a los caracteres que pueden aparecer en el valor.
- Cómo comparar valores, es decir, cuándo dos valores pueden ser "iguales", esto pudiera no ser cuando son idénticos.

Una limitación importante de las definiciones de atributos (con respecto a las propiedades) es que no es posible controlar los valores que se han de aceptar para ese atributo.

Las definiciones de atributo son independientes de las definiciones de clases de objeto. Cualquier atributo puede ser utilizado con cualquier definición de clase de objeto. Se permite la subtipificación de valores de propiedades.

Los métodos utilizados para asignar los OID ayudan a asegurar que un determinado OID no identifica dos tipos diferentes (tipos de servicio, tipos de interfaz, tipos de propiedades de servicio) o propiedades (atributos de propiedades), incluso entre múltiples intermediarios poseídos por diferentes organizaciones. Nunca debe haber un conflicto porque dos intermediarios diferentes asignaron el mismo OID a dos tipos o propiedades diferentes.

Las definiciones de tipos y propiedades son conocidas por el DSA que almacena la base de datos del intermediario y el T-DUA. En un DSA, el esquema X.500 está almacenado como parte de la base de datos X.500 y, en teoría, puede ser gestionado dinámicamente.

9 Propiedades dinámicas

La utilización de la gestión del esquema X.500 para realizar parte del depositario de tipos tiene repercusiones en cuanto al almacenamiento de propiedades dentro del esquema X.500. La gestión de este esquema rechaza el intento de crear un asiento que no tenga los atributos obligatorios (propiedades), aun cuando puedan ser propiedades dinámicas.

La solución adoptada es que el T-DUA añade cualesquiera propiedades dinámicas a la oferta de servicio antes de almacenarla en el directorio X.500. Una propiedad dinámica en un exportador está "marcada" por la utilización de la estructura de propiedad dinámica en vez de un valor normal para el valor "cualquiera" de una propiedad. Un intermediario que admite propiedades dinámicas reconoce esta estructura y la acepta como una propiedad con un valor dinámico. El T-DUA almacena una propiedad dinámica con un valor "ficticio" y crea, al mismo tiempo, un "valor de propiedad dinámica" asociado en el atributo propiedades dinámicas que contiene la información necesaria para que el T-DUA obtenga un valor de propiedad correctamente tipificado, cuando proceda. El valor "ficticio" del atributo de propiedad correspondiente es omitido durante la comparación de una oferta de servicio. Se define un "valor de propiedad dinámica" separado para cada posible propiedad dinámica que ha de ser almacenada en el intermediario.

Por ejemplo, se supone que un exportador exporta una oferta de servicio con la propiedad "longitud de cola" marcada como "dinámica". Al crear el asiento de oferta de servicio en la base de datos X.500, el T-DUA crea un caso de "longitud de cola" y el "valor de propiedad dinámica longitud de cola" en ese asiento. Cuando se lee ese asiento de oferta de servicio, la presencia del "valor de propiedad dinámica longitud de cola" en el atributo propiedades dinámicas indica que el valor del atributo (propiedad) correspondiente "longitud de cola" hallado en la base de datos X.500 se ha de omitir y se debe obtener el valor real de la interfaz almacenada en el valor de propiedad dinámica.

Otro método sería incluir la conmutación "dinámica" como parte de cada definición de atributo (es decir, cualquier atributo tiene un valor real o el conjunto de opciones "dinámicas"). Este método se ha rechazado porque complicaría todas las definiciones de atributos y el procesamiento.

9.1 Exportación de una oferta de servicio

Al exportar una oferta de servicio, el cliente especifica el OID del tipo de servicio y un conjunto de propiedades (atributos). Si el intermediario admite propiedades dinámicas y hay propiedades dinámicas indicadas en la oferta de servicio, el T-DUA construye valores "ficticios" y el atributo propiedades dinámicas con un valor de propiedad dinámica para cada propiedad dinámica en la oferta de servicio. El T-DUA fija también el atributo "tiene propiedades dinámicas"

a "verdadero". Si el intermediario no admite propiedades dinámicas, se rechaza cualquier oferta de servicio con propiedades dinámicas. El directorio X.500 rechaza también la exportación si la oferta de servicio no concuerda con el esquema.

Si el cliente desea exportar una oferta de un nuevo tipo de servicio, la exportación se convierte en un proceso en dos etapas. Primero, cualesquiera nuevas definiciones de "tipo" y cualesquiera nuevas definiciones de "propiedad" se añaden al esquema X.500 mantenido por el depositario de tipos. Una vez que los nuevos tipos y atributos (propiedades) han sido registrados en el DSA, el cliente puede exportar la nueva oferta de servicio.

9.2 Importación de una oferta de servicio

Al importar una oferta de servicio, el cliente especifica el tipo de servicio (OID) que desea encontrar y una "constricción de concordancia".

En la Recomendación X.500 se dispone de una forma limitada de subclasificación. Una clase de objeto puede ser definido como una "subclase" de una o más clases de objetos existentes. En este caso, la especificación del tipo de propiedades de servicio (OID) de una superclase seleccionará también ofertas de servicio de la subclase. La limitación del concepto superclase/subclase está en las listas de "debe contener" y "puede contener" de la definición de clase de objeto. Una subclase hereda las listas "debe contener" y "puede contener" de sus superclases y sólo puede añadir atributos adicionales.

El T-DUA busca la base de datos X.500 para comparar las ofertas de servicio y las ofertas por procuración (si procede), primero en cuanto al tipo de servicio requerido y después para comparar las ofertas con la constricción y el tipo requeridos.

Si el intermediario admite propiedades dinámicas y el importador pide la utilización de propiedades dinámicas, el valor del atributo "tiene propiedades dinámicas" se solicita también en la primera búsqueda. Si este atributo es "verdadero", el filtro de constricciones de concordancia se escribe de nuevo para continuar. Para cada asiento devuelto, el T-DUA revisa la lista de atributos devueltos. Si está presente algún "valor de propiedad dinámica", el T-DUA pide el valor "real" de la propiedad asociada de la interfaz contenida en el valor de propiedad dinámica. Una vez devueltas las propiedades dinámicas, el T-DUA debe completar la evaluación de la constricción de concordancia.

Anexo A

Definición esquemática de definiciones de intermediario

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

```

TraderDefinitions {joint-iso-itu-t 2}
DEFINITIONS ::=
BEGIN
IMPORTS
    informationFramework, selectedAttributeTypes, authenticationFramework
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                                usefulDefinitions(0) 2}

    CONTENT-RULE, NAME-FORM, STRUCTURE-RULE, OBJECT-CLASS, MATCHING-RULE,
    ATTRIBUTE, top, ObjectClassKind, objectIdentifierMatch,
    DistinguishedName
        FROM InformationFramework informationFramework

    DirectoryString {}, commonName, description, presentationAddress,
    distinguishedName, caseIgnoreMatch, caseIgnoreSubstringsMatch,
    caseExactMatch, booleanMatch, integerMatch, octetStringMatch
        FROM SelectedAttributeTypes selectedAttributeTypes

    userPassword
        FROM AuthenticationFramework authenticationFramework ;

-- Trader Entry
traderEntry OBJECT-CLASS ::= {
    SUBCLASS OF          { top }
    MUST CONTAIN         {commonName | traderInterface | dsaName |
                          typeRepos | defSearchCard | maxSearchCard |
                          defMatchCard | maxMatchCard | defReturnCard |
                          maxReturnCard | defHopCount | maxHopCount |
                          defFollowPolicy | maxFollowPolicy |
                          maxLinkFollowPolicy | supportsModifiableProperties |
                          supportsDynamicProperties | supportsProxyOffers |
                          maxList | requestIdStem}
    MAY CONTAIN          {description | userPassword}
    ID                   id-trader-oc-traderEntry}

traderInterface ATTRIBUTE ::= {
    SUBTYPE OF           presentationAddress
    SINGLE VALUE         TRUE
    ID                   id-trader-at-traderInterface}

```

```

dsaName ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-dsaName}

typeRepos ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-typeRepos}

defSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defSearchCard }

maxSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxSearchCard}

defMatchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defMatchCard}

maxMatchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxMatchCard}

defReturnCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defReturnCard}

maxReturnCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxReturnCard}

```

ISO/CEI 13235-3 : 1998 (S)

```
defHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defHopCount}

maxHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxHopCount}

defFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defFollowPolicy }

FollowOption ::= ENUMERATED{
    localOnly    (0),
    ifNoLocal    (1),
    always       (2)}

maxFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxFollowPolicy }

maxLinkFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxLinkFollowPolicy }

supportsModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsModifiableProperties}

supportsDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsDynamicProperties}
```

```

supportsProxyOffers ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsProxyOffers}

maxList ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxList }

requestIdStem ATTRIBUTE ::= {
    WITH SYNTAX                OCTET STRING (SIZE (0..ub-request-id-stem))
    EQUALITY MATCHING RULE     octetStringMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-requestIdStem }

--Trader Policy Entry

traderPolicyEntryNF NAME-FORM ::= {
    NAMES                       traderPolicyEntry
    WITH ATTRIBUTES             {commonName}
    ID                           id-trader-nf-traderPolicy}

traderPolicyEntry OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    MUST CONTAIN                 {commonName }
    MAY CONTAIN                 {typeManagementConstraint | searchConstraint |
                                offerAcceptanceConstraint }
    ID                           id-trader-oc-traderPolicy}

PolicySpecification ::= CHOICE {
    stringRule                   [0]   DirectoryString{ub-policy-string-rule},
    policyObjectId               [1]   DistinguishedName }

policySpecificationMatch MATCHING-RULE ::= {
    SYNTAX                       PolicySpecification
    ID                           id-trader-mr-policySpecificationMatch}

-- The rule returns TRUE if two specifications contain exactly
-- the same characters.

typeManagementConstraint ATTRIBUTE ::= {
    WITH SYNTAX                  PolicySpecification
    EQUALITY MATCHING RULE       policySpecificationMatch
    SINGLE VALUE                 TRUE
    ID                           id-trader-at-typeManagementConstraint}

```

ISO/CEI 13235-3 : 1998 (S)

```
searchConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-searchConstraint}

offerAcceptanceConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-offerAcceptanceConstraint}

-- Service Offer Entry

serviceOfferEntryNF NAME-FORM ::= {
    NAMES                      serviceOfferEntry
    WITH ATTRIBUTES            {sOfferId}
    ID                          id-trader-nf-serviceOffer}

serviceOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    MUST CONTAIN                {sOfferId | serviceInterfaceId | serviceTypeId
                                hasDynamicProperties | hasModifiableProperties }
    MAY CONTAIN                {dynamicProps}
    ID                          id-trader-oc-serviceOffer}

sOfferId ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString{ub-s-offer-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-sOfferId}

serviceInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    SINGLE VALUE               TRUE
    ID                          id-trader-at-serviceInterfaceId}

serviceTypeId ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE     objectIdentifierMatch
    ID                          id-trader-at-serviceTypeId }

hasDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-hasDynamicProperties}
```

```

hasModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-hasModifiableProperties}

dynamicProps ATTRIBUTE ::= {
    WITH SYNTAX                SEQUENCE OF DynamicPropValue
    ID                          id-trader-at-dynamicProps}

DynamicPropValue ::= SEQUENCE {
    propertyType                OBJECT IDENTIFIER,
    dynamicPropEvalIf          DistinguishedName,
    extraInfo                  DirectoryString{ub-dynamic-value-extra-info}}

dynamicPropValueMatch MATCHING-RULE ::= {
    SYNTAX                      DynamicPropValue
    ID                          Id-trader-mr-dynamicPropValueMatch}

-- The rule returns TRUE if two values contain exactly the same characters.

-- Trader Link Entry

traderLinkEntryNF NAME-FORM ::= {
    NAMES                       traderLinkEntry
    WITH ATTRIBUTES             {linkId}
    ID                          id-trader-nf-traderLink}

traderLinkEntry OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    MUST CONTAIN                 {linkName | linkId | targetTraderInterfaceId |
                                defPassOnFollowRule | limitingFollowRule}
    ID                          id-trader-oc-traderLink}

linkName ATTRIBUTE ::= {
    WITH SYNTAX                 DirectoryString {ub-link-name}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-linkName}

linkId ATTRIBUTE ::= {
    WITH SYNTAX                 DirectoryString {ub-link-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-linkId}

```

ISO/CEI 13235-3 : 1998 (S)

```
targetTraderInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-targetTraderInterfaceId}

defPassOnFollowRule ATTRIBUTE ::= {
    WITH SYNTAX         FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defPassOnFollowRule }

limitingFollowRule ATTRIBUTE ::= {
    WITH SYNTAX         FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-limitingFollowRule }

-- Proxy Offer Entry

proxyOfferEntryNF NAME-FORM ::= {
    NAMES                proxyOfferEntry
    WITH ATTRIBUTES      {proxyOfferId}
    ID                  id-trader-nf-proxyOffer}

proxyOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {proxyOfferId | proxyLookUpInterfaceId |
                        hasDynamicProperties | hasModifiableProperties |
                        ifMatchAll | constraintRecipe}
    MAY CONTAIN          { dynamicProps }
    ID                  id-trader-oc-proxyOffer}

proxyOfferId ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-proxy-offer-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyOfferId}

proxyLookUpInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyLookUpInterfaceId}
```

```
constraintRecipe ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-constraint-recipe}
    EQUALITY MATCHING RULE  caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE          TRUE
    ID                    id-trader-at-constraintRecipe}
```

```
ifMatchAll ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE  booleanMatch
    SINGLE VALUE          TRUE
    ID                    id-trader-at-ifMatchAll }
```

-- Interface Entry

```
interfaceEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    KIND                  auxiliary
    MUST CONTAIN          {interfaceReference | interfaceType}
    ID                    id-trader-oc-interfaceEntry}
```

```
interfaceReference ATTRIBUTE ::= {
    WITH SYNTAX          InterfaceId
    EQUALITY MATCHING RULE  caseExactMatch
    ID                    id-trader-at-interfaceReference}
```

```
InterfaceId ::=          DirectoryString {ub-interface-id}
```

```
interfaceType ATTRIBUTE ::= {
    WITH SYNTAX          InterfaceTypeName
    EQUALITY MATCHING RULE  objectIdentifierMatch
    ID                    id-trader-at-interfaceType}
```

```
InterfaceTypeName ::=    OBJECT IDENTIFIER
```

-- Object Identifier Assignments

```
id-trader                OBJECT IDENTIFIER ::= {joint-iso-itu-t
trader(100)}
```

```
id-trader-at             OBJECT IDENTIFIER ::= {id-trader 4}
```

```
id-trader-oc             OBJECT IDENTIFIER ::= {id-trader 6}
```

```
id-trader-mr             OBJECT IDENTIFIER ::= {id-trader 13}
```

```
id-trader-nf             OBJECT IDENTIFIER ::= {id-trader 15}
```

```
id-trader-oc-traderEntry OBJECT IDENTIFIER ::= {id-trader-oc 0}
```

```
id-trader-oc-serviceOffer OBJECT IDENTIFIER ::= {id-trader-oc 1}
```

```
id-trader-oc-proxyOffer  OBJECT IDENTIFIER ::= {id-trader-oc 2}
```

ISO/CEI 13235-3 : 1998 (S)

id-trader-oc-traderLink	OBJECT IDENTIFIER ::= {id-trader-oc 3}
id-trader-oc-traderPolicy	OBJECT IDENTIFIER ::= {id-trader-oc 4}
id-trader-oc-interfaceEntry	OBJECT IDENTIFIER ::= {id-trader-oc 5}
id-trader-nf-serviceOffer	OBJECT IDENTIFIER ::= {id-trader-nf 1}
id-trader-nf-traderLink	OBJECT IDENTIFIER ::= {id-trader-nf 2}
id-trader-nf-traderPolicy	OBJECT IDENTIFIER ::= {id-trader-nf 3}
id-trader-nf-proxyOffer	OBJECT IDENTIFIER ::= {id-trader-nf 4}
id-trader-at-traderInterface	OBJECT IDENTIFIER ::= {id-trader-at 0}
id-trader-at-typeRepos	OBJECT IDENTIFIER ::= {id-trader-at 1}
id-trader-at-defSearchCard	OBJECT IDENTIFIER ::= {id-trader-at 2}
id-trader-at-maxSearchCard	OBJECT IDENTIFIER ::= {id-trader-at 3}
id-trader-at-defMatchCard	OBJECT IDENTIFIER ::= {id-trader-at 4}
id-trader-at-maxMatchCard	OBJECT IDENTIFIER ::= {id-trader-at 5}
id-trader-at-commonName	OBJECT IDENTIFIER ::= {id-trader-at 6}
id-trader-at-dsaName	OBJECT IDENTIFIER ::= {id-trader-at 7}
id-trader-at-defReturnCard	OBJECT IDENTIFIER ::= {id-trader-at 10}
id-trader-at-maxReturnCard	OBJECT IDENTIFIER ::= {id-trader-at 11}
id-trader-at-defHopCount	OBJECT IDENTIFIER ::= {id-trader-at 12}
id-trader-at-maxHopCount	OBJECT IDENTIFIER ::= {id-trader-at 13}
id-trader-at-defFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 14}
id-trader-at-maxLinkFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 15}
id-trader-at-maxFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 16}
id-trader-at-supportsModifiableProperties	OBJECT IDENTIFIER ::= {id-trader-at 20}
id-trader-at-supportsDynamicProperties	OBJECT IDENTIFIER ::= {id-trader-at 21}
id-trader-at-supportsProxyOffers	OBJECT IDENTIFIER ::= {id-trader-at 22}
id-trader-at-maxList	OBJECT IDENTIFIER ::= {id-trader-at 23}
id-trader-at-requestIdStem	OBJECT IDENTIFIER ::= {id-trader-at 24}
id-trader-at-typeManagementConstraint	OBJECT IDENTIFIER ::= {id-trader-at 25}
id-trader-at-searchConstraint	OBJECT IDENTIFIER ::= {id-trader-at 30}
id-trader-at-offerAcceptanceConstraint	OBJECT IDENTIFIER ::= {id-trader-at 31}
id-trader-at-sOfferId	OBJECT IDENTIFIER ::= {id-trader-at 32}
id-trader-at-serviceTypeId	OBJECT IDENTIFIER ::= {id-trader-at 33}
id-trader-at-serviceInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 34}
id-trader-at-hasDynamicProperties	OBJECT IDENTIFIER ::= {id-trader-at 35}
id-trader-at-hasModifiableProperties	OBJECT IDENTIFIER ::= {id-trader-at 40}
id-trader-at-dynamicProps	OBJECT IDENTIFIER ::= {id-trader-at 41}
id-trader-at-linkId	OBJECT IDENTIFIER ::= {id-trader-at 42}
id-trader-at-linkName	OBJECT IDENTIFIER ::= {id-trader-at 43}
id-trader-at-targetTraderInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 44}
id-trader-at-defPassOnFollowRule	OBJECT IDENTIFIER ::= {id-trader-at 45}
id-trader-at-limitingFollowRule	OBJECT IDENTIFIER ::= {id-trader-at 50}
id-trader-at-proxyOfferId	OBJECT IDENTIFIER ::= {id-trader-at 51}
id-trader-at-proxyLookUpInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 52}
id-trader-at-constraintRecipe	OBJECT IDENTIFIER ::= {id-trader-at 53}

id-trader-at-ifMatchAll	OBJECT IDENTIFIER ::= {id-trader-at 55}
id-trader-at-interfaceReference	OBJECT IDENTIFIER ::= {id-trader-at 60}
id-trader-at-interfaceType	OBJECT IDENTIFIER ::= {id-trader-at 61}
id-trader-mr-policySpecificationMatch	OBJECT IDENTIFIER ::= {id-trader-mr 1}
id-trader-mr-dynamicPropValueMatch	OBJECT IDENTIFIER ::= {id-trader-mr 2}

-- Upperbounds

ub-common-name	INTEGER ::= 64
ub-request-id-stem	INTEGER ::= 1024
ub-policy-string-rule	INTEGER ::= 1024
ub-s-offer-id	INTEGER ::= 64
ub-dynamic-value-extra-info	INTEGER ::= 1024
ub-link-name	INTEGER ::= 64
ub-link-id	INTEGER ::= 64
ub-proxy-offer-id	INTEGER ::= 64
ub-constraint-recipe	INTEGER ::= 1024
ub-interface-id	INTEGER ::= 1024

END

Anexo B

Ejemplo de definición esquemática de descripción de servicio

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

```

PrinterServiceOfferDefinitions {joint-iso-itu-t 2}
DEFINITIONS ::=
BEGIN
IMPORTS
    informationFramework, selectedAttributeTypes
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) modules(1)
            usefulDefinitions(0) 2}

OBJECT-CLASS, MATCHING-RULE, ATTRIBUTE, top, ObjectClassKind, DistinguishedName
    FROM InformationFramework informationFramework

DirectoryString {}, caseIgnoreMatch, caseIgnoreSubstringsMatch, caseExactMatch,
booleanMatch, integerMatch, integerOrderMatch
    FROM SelectedAttributeTypes selectedAttributeTypes

id-trader-at, id-trader-oc-serviceOffer
    FROM id-trader{joint-iso-itu-t trader(100)} ;

printerServiceOffer OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    KIND                  auxiliary
    MUST CONTAIN         printerType}
MAY CONTAIN             {locationRoom | locationBuilding | costPerPage |
    languagesSupported | pagesPerMinute | pageSize |
    dotsPerInch | colourCapable | driverName | queueLength}
    ID                   id-trader-oc-serviceOffer-printer}

printerType ATTRIBUTE ::= {
    WITH SYNTAX           DirectoryString {ub-trader-so-printerType}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE         TRUE
    ID                   id-trader-at-so-printerType}

locationRoom ATTRIBUTE ::= {
    WITH SYNTAX           DirectoryString {ub-trader-so-locationRoom}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE         TRUE
    ID                   id-trader-at-so-locationRoom }

```

```
locationBuilding ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-locationBlg}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-locationBlg}
```

```
costPerPage ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-costPerPage}
```

```
languagesSupported ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-langSupp}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-langSupp}
```

```
pagesPerMinute ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-pagesPerMinute}
```

```
pagesPerMinute ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-pagesPerMinute}
```

```
dotsPerInch ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-dotsPerInch}
```

ISO/CEI 13235-3 : 1998 (S)

```
colourCapable ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-colourCapable}

driverName ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-driverName}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-driverName}

queueLength ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE     integerOrderMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-queueLength }

-- Object Identifiers

id-trader-oc-serviceOffer-printer
                                OBJECT IDENTIFIER ::= {id-trader-oc-serviceOffer 0}

id-trader-at-so                 OBJECT IDENTIFIER ::= {id-trader-at 100}
id-trader-at-so-printerType     OBJECT IDENTIFIER ::= {id-trader-at-so 0}
id-trader-at-so-locationRoom    OBJECT IDENTIFIER ::= {id-trader-at-so 1}
id-trader-at-so-locationBlg     OBJECT IDENTIFIER ::= {id-trader-at-so 2}
id-trader-at-so-costPerPage     OBJECT IDENTIFIER ::= {id-trader-at-so 3}
id-trader-at-so-langSupp        OBJECT IDENTIFIER ::= {id-trader-at-so 4}
id-trader-at-so-pagesPerMinute  OBJECT IDENTIFIER ::= {id-trader-at-so 5}
id-trader-at-so-pageSize        OBJECT IDENTIFIER ::= {id-trader-at-so 6}
id-trader-at-so-dotsPerInch     OBJECT IDENTIFIER ::= {id-trader-at-so 7}
id-trader-at-so-colourCapable   OBJECT IDENTIFIER ::= {id-trader-at-so 8}
id-trader-at-so-driverName      OBJECT IDENTIFIER ::= {id-trader-at-so 9}
id-trader-at-so-queueLength     OBJECT IDENTIFIER ::= {id-trader-at-so 10}

-- Upperbounds

ub-trader-so-printerType        INTEGER ::= 64
ub-trader-so-locationRoom       INTEGER ::= 64
ub-trader-so-locationBlg        INTEGER ::= 64
ub-trader-so-langSupp           INTEGER ::= 64
ub-trader-so-pageSize           INTEGER ::= 64
ub-trader-so-driverName         INTEGER ::= 64
```

END

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
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes de programación