



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.781

(08/2001)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Gestión de interconexión de sistemas abiertos –
Funciones de gestión y funciones de arquitectura de
gestión distribuida abierta

**Requisitos y directrices para los formularios de
declaración de conformidad de implementación
asociados con sistemas basados en
arquitectura de intermediario de petición de
objeto común**

Recomendación UIT-T X.781

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

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfases	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.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.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.

Recomendación UIT-T X.781

Requisitos y directrices para los formularios de declaración de conformidad de implementación asociados con sistemas basados en arquitectura de intermediario de petición de objeto común

Resumen

Esta Recomendación especifica el formulario de declaración de conformidad de implementación (ICS) basado en interfaces de sistema del tipo CORBA que, serán utilizados para la gestión de redes de telecomunicación, define además la metodología de prueba a utilizar para las interfaces basadas en CORBA.

Orígenes

La Recomendación UIT-T X.781, preparada por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 13 de agosto de 2001.

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 Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), 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 1 de la AMNT.

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 2002

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	1
2	Referencias.....	1
3	Definiciones	1
3.1	Definiciones relativas a las declaraciones de conformidad de implementaciones	1
4	Abreviaturas y acrónimos	2
5	Bases de la metodología de pruebas de conformidad para interfaces de sistemas de gestión basadas en CORBA.....	2
5.1	Visión general	2
5.2	Metodología para las pruebas de conformidad de interfaces basadas en CORBA.....	2
6	Formulario ICS para interfaz de gestión basado en CORBA	3
6.1	Visión general del IDL	4
6.2	Requisitos y directrices para la especificación de formularios CIICS.....	4
6.2.1	Instrucciones generales para la especificación de formulario CIICS	5
6.2.2	Formulario de soporte de interfaz.....	6
6.2.3	Formulario de soporte de atributo	6
6.2.4	Formulario de soporte de operación	7
6.2.5	Formulario de soporte de tipo de datos	8
7	Instrucciones para rellenar el formulario CIICS.....	9
7.1	Definición de soportado.....	9
7.2	La columna Status.....	9
7.3	La columna Soporte	10
7.4	La columna "Constricciones y valores"	10
7.5	La columna "Información adicional"	11
7.6	La columna "Índice"	12
7.7	La columna de "Subíndice"	12
7.8	La columna "Categoría".....	12
7.9	La columna "Propiedad de campo".....	12
7.10	Las columnas "Identificador de interfaz", "Identificador de atributo", "Identificador de operación"	12
7.11	La columna "Interfaz derivada"	13
Anexo A	– Ejemplo de especificación de formulario CIICS.....	13
A.1	Definición en CORBA IDL	13
A.2	CIICS	13

	Página
Anexo B – Ejemplo de una especificación de "Formulario de soporte de tipo de datos"	16
B.1 Definición IDL CORBA.....	16
B.2 CIICS	17
Anexo C – Ejemplo que muestra la especificación del formulario "Appendix any type support proforma"	19
C.1 Definición CORBA IDL.....	19
C.2 CIICS	20
Anexo D – Ejemplo que muestra la especificación de "formulario de soporte valuetype"	21
D.1 Definición CORBA IDL.....	21
D.2 CIICS	22

Recomendación UIT-T X.781

Requisitos y directrices para los formularios de declaración de conformidad de implementación asociados con sistemas basados en arquitectura de intermediario de petición de objeto común

1 Alcance

Esta Recomendación contiene los requisitos y directrices para el formulario de declaración de conformidad de implementación de interfaz CORBA (CIICS, *CORBA-based interface implementation conformance statement*) y la especificación de este formulario. La CIICS es una declaración, hecha por un implementador, por la que pretende la conformidad con una definición de interfaz basada en CORBA/IDL.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- UIT-T X.296 (1995), *Metodología y marco de las pruebas de conformidad de interconexión de sistemas abiertos de las Recomendaciones sobre los protocolos para aplicaciones del UIT-T – Declaraciones de conformidad de implementación.*
- UIT-T X.724 (1996), *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Requisitos y directrices para los formularios de declaración de conformidad de implementación asociados con la gestión de interconexión de sistemas abiertos.*
- ISO/CEI 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation conformance Statements.*
- OMG (1998), *The Common Object Request Broker: Architecture and Specification, Revision 2.3.*

3 Definiciones

3.1 Definiciones relativas a las declaraciones de conformidad de implementaciones

En esta Recomendación se utilizan los siguientes términos definidos en UIT-T X.296 e ISO/CEI 9646-7:

- a) ítem (ICS);
- b) pregunta (ICS);
- c) status (valor);
- d) respuesta (soporte).

4 Abreviaturas y acrónimos

En esta Recomendación se utilizan las siguientes siglas.

CIICS	Declaración de conformidad de implementación de interfaz basada en CORBA (<i>CORBA-based interface implementation conformance statement</i>)
CORBA	Arquitectura de intermediario de petición de objeto común (<i>common object request broker architecture</i>)
ICS	Declaración de conformidad de implementación (<i>implementation conformance statement</i>)
IDL	Lenguaje de definición de interfaz (<i>interface definition language</i>)
IUT	Implementación sometida a prueba (<i>implementation under test</i>)
IXIT	Información suplementaria de implementación para pruebas (<i>implementation extra information for testing</i>)
ODP	Procesamiento distribuido abierto (<i>open distributed processing</i>)

5 Bases de la metodología de pruebas de conformidad para interfaces de sistemas de gestión basadas en CORBA

5.1 Visión general

La conformidad relaciona una implementación con una norma. Da una indicación de la manera en la cual sistemas, implementados con respecto a una norma, pueden variar sin que se produzcan errores en su cooperación. Cuando una implementación cumple estos requisitos, es conforme con la norma. La comprobación de las declaraciones se hace mediante la prueba de conformidad. El punto de partida consiste en definir los requisitos de conformidad en especificaciones de interfaz independientes de la implementación, sobre la base de una identificación de puntos de referencia. Una especificación de interfaz de gestión debe definir los puntos de referencia de conformidad, en los que se debe probar un objeto, para determinar si cumple un conjunto de criterios de conformidad. Durante la prueba se evalúa y observa un cierto número de estímulos y eventos en estos puntos de conformidad.

Las especificaciones de interfaz de gestión deben incluir declaraciones de conformidad que identifiquen los puntos de referencia de conformidad en cada interfaz de los objetos especificados.

Puesto que el flujo de información entre dos componentes de sistema se efectúa, en general, a través de varios puntos de referencia, la prueba de conformidad debe tener en cuenta:

- a) la prueba de flujo de información en cada punto de referencia; y
- b) la prueba de coherencia entre las combinaciones.

Por tanto, es necesario efectuar una prueba coordinada en todos los puntos de referencia identificados.

5.2 Metodología para las pruebas de conformidad de interfaces basadas en CORBA

Los requisitos de las pruebas de conformidad pueden ser estáticos o dinámicos:

- a) Los requisitos de pruebas de conformidad estáticos indican la funcionalidad mínima necesaria para estas pruebas. La base de la prueba de conformidad estática es el llamado formulario ICS, en el cual se definen las limitaciones y posibilidades funcionales de la norma. Este documento tiene la forma de un cuestionario, al que debe responder el implementador.

- b) Los requisitos de pruebas de conformidad dinámicos especifican todos los posibles comportamientos de una implementación visibles en los puntos de referencia identificados. La prueba de conformidad dinámica incluye la realización de casos de prueba en un sistema de prueba.

Los casos de prueba pueden ser casos de prueba concretos y casos de prueba abstractos. Los casos de prueba concretos se derivan de casos de prueba abstractos que deben ser presentados o definidos por la norma. Una precondition básica para la derivación de los casos de prueba, es la definición de los propósitos de la prueba por la norma.

Antes de iniciar la prueba en un laboratorio se deben seguir los siguientes pasos:

- El implementador debe rellenar el correspondiente formulario ICS. El IXIT (información suplementaria de implementación para pruebas) contiene información que hace más precisas las declaraciones del formulario ICS, o que son de interés para la realización de la prueba.
- Los llamados casos de prueba reales (concretos) se deben derivar de los casos de prueba abstractos de la norma, o directamente de los propósitos de prueba, en los cuales se realizan cosas genéricas (tipos, etc.) según sea la implementación.

En la figura 5-1 se presenta este enfoque.

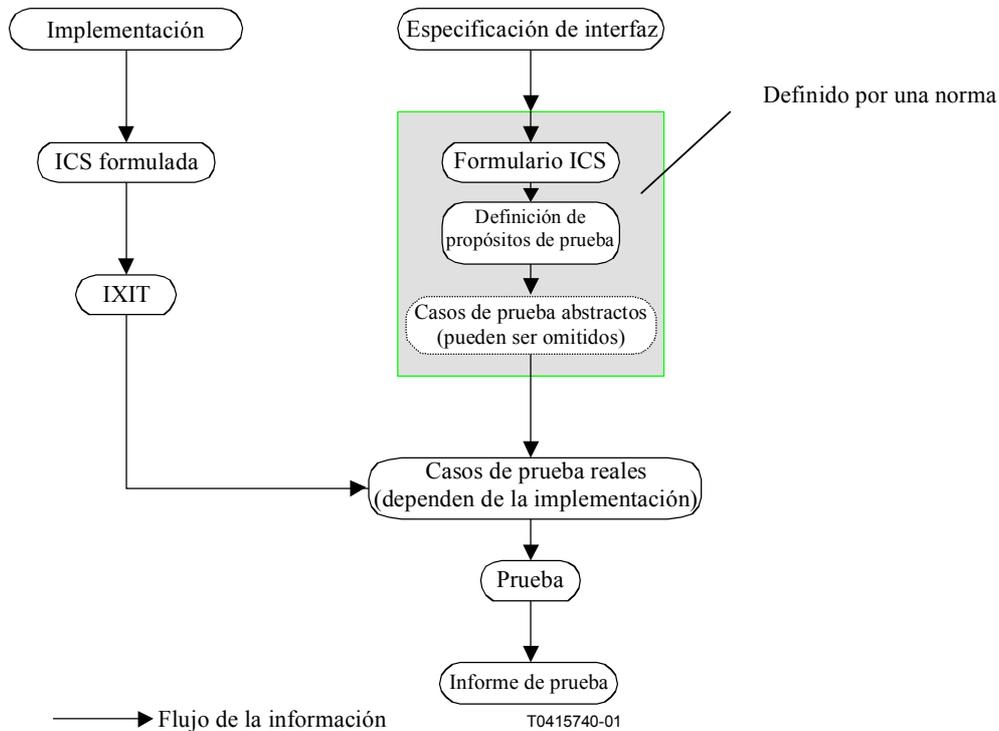


Figura 5-1/X.781 – Proceso de pruebas de interfaces basadas en CORBA

6 Formulario ICS para interfaz de gestión basado en CORBA

El lenguaje de definición de interfaz (IDL, *interface definition language*) CORBA (brevemente, CORBA IDL) se utiliza para definir interfaces de objetos en sistemas basados en CORBA. El formulario CIICS debe incluir las características del IDL, tales como los módulos IDL, interfaces y otros tipos de IDL. En esta cláusula se presentan las características de CORBA IDL, y luego se especifican los formularios CIICS de acuerdo con estas características de lenguaje.

6.1 Visión general del IDL

El lenguaje de definición de interfaz (IDL) del OMG (grupo de gestión de objeto) es el lenguaje utilizado para describir las interfaces llamadas por objetos cliente y proporcionadas por implementaciones de objeto. Una definición de interfaz escrita en OMG IDL define completamente la interfaz y especifica los parámetros de cada operación. Una implementación sometida a prueba (IUT, *implementation under test*) puede incluir una función cliente de interfaz o una función servidor de interfaz. Por tanto, en el formulario de interfaz debe existir un espacio disponible para especificar el lado de la función (cliente o servidor) que la IUT implementa.

La gramática del OMG IDL es muy similar a la norma ANSI C++ propuesta. Tiene construcciones tales como module (módulo), interface (interfaz), constant (constante), type (tipo), operation (operación), attribute (atributo), exception (excepción), valuetype (tipo de valor), state member (miembro de estado) y declaraciones de contexto. De acuerdo con la especificación de sintaxis IDL, la relación de contención entre estas construcciones se puede ilustrar como se indica en la figura 6-1.

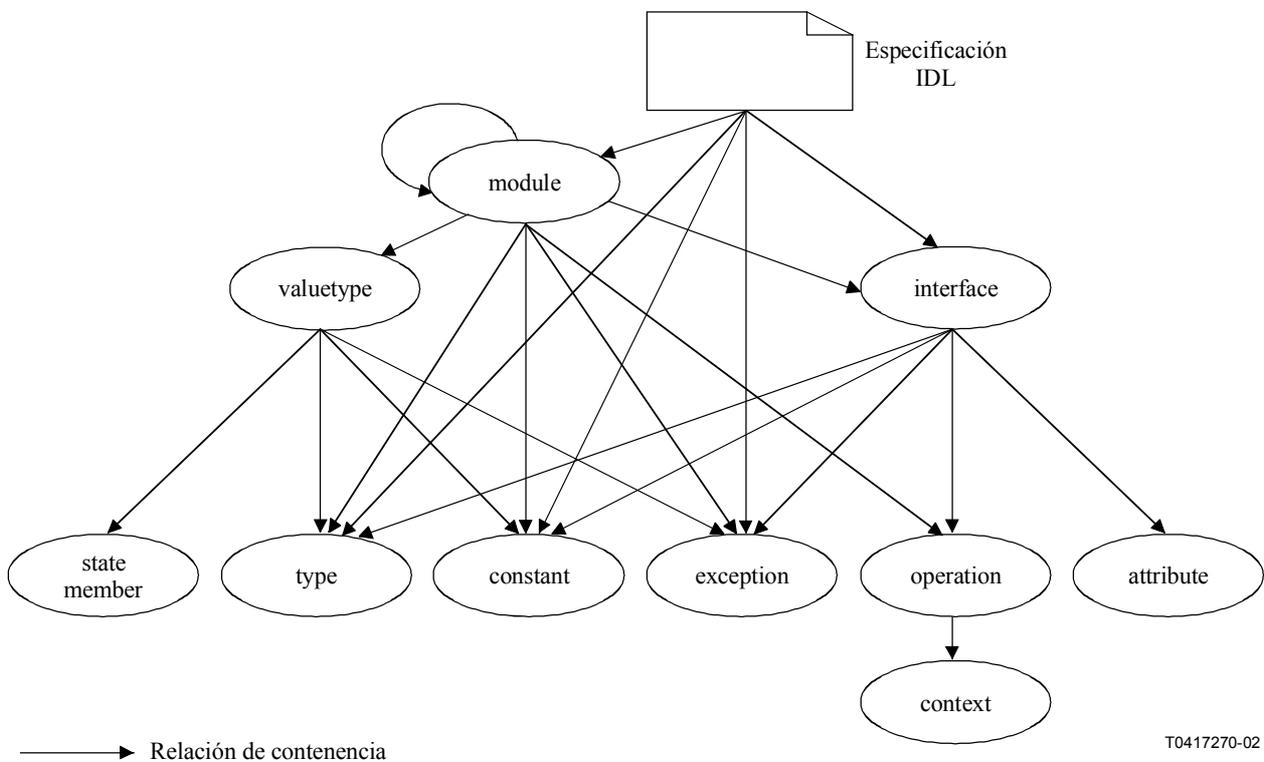


Figura 6-1/X.781 – Relación de contención en la especificación IDL

Los formularios ICS representarán las relaciones indicadas en esta figura.

6.2 Requisitos y directrices para la especificación de formularios CIICS

Las especificaciones de formulario seguirán el estilo que se presenta en las cláusulas siguientes y proporcionarán la información requerida en esta Recomendación. Se pueden incluir cuadros adicionales siempre y cuando sean necesarios para suministrar más información.

Hay tres niveles de documentación pertinente al CIICS, a saber:

- a) directrices o herramientas proporcionadas por una Recomendación para la producción de los formularios CIICS;
- b) un formulario CIICS, asociado con una norma relacionada con la gestión de red basada en CORBA, que debe ser rellanado por un suministrador de la implementación, y que, una vez rellanado, se convierte en una CIICS;
- c) una CIICS preparada por un suministrador de la implementación como parte de una reivindicación de la conformidad con una norma relacionada con la gestión de red basada en CORBA.

6.2.1 Instrucciones generales para la especificación de formulario CIICS

En la presente Recomendación se proporcionan instrucciones para construir especificaciones de formularios ICS para sistemas de gestión basados en CORBA. Los formularios CIICS incluyen cuatro formularios de soporte: formulario de soporte de interfaz, formulario de soporte de atributo, formulario de soporte de operación y formulario de soporte de tipos de datos. Todos ellos se presentan en forma de cuadros, similares a los formularios especificados en UIT-T X.724.

La cláusula 6.2.2 describe el formulario CIICS para interfaces. La cláusula 6.2.3 describe el formulario CIICS para atributos. La cláusula 6.2.4 describe el formulario CIICS para operaciones. La cláusula 6.2.5 describe el formulario CIICS para tipos de datos complejos. Los anexos A, B, C y D proporcionan ejemplos de especificación de formulario CIICS, que deben ser rellanados por el suministrador de una implementación.

Las notaciones de uso común que se citan a continuación, y que están definidas en UIT-T X.291 e ISO/CEI 9646-2, y en UIT-T X.296 e ISO/CEI 9646-7, se utilizan para la columna "Status" en la presente Recomendación:

- m Obligatorio (*Mandatory*)
- o Facultativo (*Optional*)
- No aplicable o fuera del alcance

NOTA – Las notaciones "m", "o" vienen precedidas de una "c": cuando están anidadas dentro de un ítem facultativo del mismo cuadro.

Las siguientes notaciones de uso común, definidas en UIT-T X.291 e ISO/CEI 9646-2, y en UIT-T X.296 e ISO/CEI 9646-7, se utilizan para la columna de respuesta de soporte:

- Y Implementada
- N No implementada
- No se requiere respuesta
- Ig Se ignora el ítem (es decir, se procesa sintácticamente pero no semánticamente)

Para la especificación de formulario CIICS se copian las cláusulas 6.2.2, 6.2.3, 6.2.4 y 6.2.5, se llenan los cuadros salvo las columnas de "Soporte" e "Información adicional", y se amplían los cuadros restantes para que cumplan los requisitos de la especificación. En el formulario CIICS se deben incluir cuadros para todos los atributos, operaciones, parámetros, excepciones y contextos definidos en el modelo de información IDL, sin importar si son derivados de superclases o añadidos por redefinición.

Para formular una CIICS a partir de un formulario CIICS, el proveedor de la implementación rellanará la columna "Soporte" y, si es necesario, las columnas "Información adicional" de todos los cuadros del formulario CIICS.

6.2.2 Formulario de soporte de interfaz

La finalidad del formulario de interfaz es proporcionar un mecanismo para que el proveedor de una implementación que pretende la conformidad con una especificación de interfaz suministre información de conformidad de una manera normalizada.

El formulario de soporte de interfaz tiene la estructura del cuadro 6-1.

Cuadro 6-1/X.781 – Cuadro de soporte de interfaz

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional

donde:

- El campo "Índice" contiene un número, que se incrementa consecutivamente a medida que se avanza en el cuadro, que permite a los lectores del CIICS referirse a cada ítem. A cada interfaz definida en un modelo de información IDL se le asigna como índice un número único.
- El campo "Identificador de interfaz" contiene el nombre absoluto de la interfaz, es decir el nombre del espacio y el nombre de la interfaz.
- El campo "Interfaz derivada" se provee para la(s) interfaz(ces) progenitora(s) directa(s) de las que se deriva la interfaz en cuestión, cuando las haya.
- Para cada interfaz ejemplificable, el campo "Status", si es obligatorio, debe ser rellenado con 'm'; si es facultativo, debe ser rellenado con 'o'. Si una interfaz no es ejemplificable, el campo "Status" debe ser rellenado con "-", lo que significa que no es aplicable.
- El campo "Información adicional" se utiliza para indicar el lado (cliente o servidor) en el que se desempeña la función de la implementación sometida a prueba (IUT)¹.

6.2.3 Formulario de soporte de atributo

El formulario para atributo tiene por finalidad proveer un mecanismo para que un suministrador de una implementación que pretende la conformidad con una especificación de atributos en una interfaz lo haga de una forma normalizada.

El formulario de soporte de atributo debe ajustarse al cuadro 6-2 y debe implementarse uno para cada interfaz IDL.

Cuadro 6-2/X.781 – Cuadro de soporte de atributo

Índice	Identificador de atributo	Constricciones y valores	Obtener		Fijar		Información adicional
			Status	Soporte	Status	Soporte	

¹ En la mayoría de los casos la IUT actúa como un servidor, pero, en casos de notificación o informe de algún evento, puede actuar como cliente y enviar información mediante la invocación de una operación proporcionada por otro servidor situado en el lado del dispositivo probador.

donde:

- El campo "Índice" es la referencia única dentro de una interfaz.
- El campo "Identificador de atributo" es el nombre relativo del atributo.
- El campo "Constricciones y valores" corresponde a la descripción del tipo de datos del atributo.
- Tanto el campo "Obtener" (Get) como "Fijar" (Set) contienen "Status" y "Soporte". La columna "Status" de "Obtener" debe ser siempre rellenada con 'm'. Si el atributo es un atributo normal, el "Status" de "Fijar" debe ser rellenado con 'm'; cuando el atributo sea del tipo "de sólo lectura", el campo "Status" se debe llenar con un '-', lo que significa que no es aplicable.
- El campo de "Información adicional" proporciona un espacio para que el implementador añada información más específica acerca de este atributo, por ejemplo, los valores por defecto cuando el objeto es creado, o el número de índice de referencia cuando el tipo de atributo es "any". Véase 7.5 para más detalles.

En el caso de atributos de tipo complejo, la información detallada acerca del tipo se amplía en un cuadro correspondiente al "Cuadro de soporte de atributo", llamado "Cuadro de soporte de tipo de datos" que se examinará en 6.2.5.

6.2.4 Formulario de soporte de operación

Este formulario tiene por finalidad suministrar un mecanismo mediante el cual un suministrador de una implementación que pretende la conformidad con una especificación de operaciones en una interfaz proporcione información de conformidad de una manera normalizada.

El formulario de soporte de operación debe ajustarse a los cuadros 6-3 y 6-4, y debe establecerse uno para cada interfaz IDL:

Cuadro 6-3/X.781 – Cuadro de soporte de operación (Parte 1)

Índice	Identificador de operación	Status	Soporte	Información adicional

donde:

El campo "Identificador de operación" es el nombre relativo de la operación.

El campo "Información adicional" proporciona un espacio para que el implementador añada información más específica acerca de la operación.

Para cada operación, si es obligatoria, el campo "Status" se debe rellenar con 'm'; si utiliza Excepciones de Sistema, lo que por lo general se especificaría no en un IDL sino en declaraciones de comportamiento asociadas, se debe insertar una explicación en el campo "Información adicional" para especificar el tipo de Excepciones de Sistema utilizadas; cuando la Excepción de Sistema NO_IMPLEMENT pueda ser utilizada para indicar que la operación no está soportada, el campo "Status" se debe rellenar con 'o'; si hay excepciones de usuario para lotes condicionales, el campo "Status" se debe rellenar con 'c'.

Las otras columnas tienen un significado similar al descrito en 6.2.2.

Cuadro 6-4/X.781 – Cuadro de soporte de atributo (Parte 2)

Índice	Subíndice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional

donde:

- El campo "Índice" se provee para el índice de cada operación que se especifica en el cuadro 6-1.
- El campo "Subíndice" describe la referencia de cada ítem relacionado con la operación, que se forma conectando el índice de operación y un número único con un separador '!'.
- El campo "Identificador de campo de operación" describe cualquier identificador que esté relacionado con la operación, incluidos los nombres de parámetro, identificadores de contexto y nombres de excepción.
- El campo "Constricciones y valores" contiene la descripción del tipo de datos de cada parámetro, valor de retorno y excepción definidos en esta operación. La información detallada será ampliada en el "formulario de soporte de tipo de datos", que se describirán en 6.2.5.
- El campo "Categoría" permite distinguir entre dos tipos diferentes de campo de operación: "parámetro" y "terminación", que se describirán en 7.8.
- El campo "Propiedad de campo" representa una descripción adicional del campo "Categoría"; describe el modo o tipos detallados del campo de operación, como se verá en 7.9.
- El campo "Información adicional" sirve para especificar alguna información precisa que deba ser declarada por el implementador. Cuando el tipo de un parámetro es "any", se debe insertar un número de índice de referencia en ese campo. Véase 7.5 e) para más detalles.

6.2.5 Formulario de soporte de tipo de datos

El formulario de soporte de tipo de datos contiene la descripción de todos los tipos de datos definidos por el usuario, que se utilizan en el modelo de información IDL. Representa una ampliación de los "Formulario de soporte de atributo" y "Formulario de soporte de operación". El "Cuadro de soporte de tipo de datos" se divide en tres subcuadros: el "subcuadro de soporte de parámetro", el "subcuadro de soporte de valor de retorno" y el "subcuadro de soporte de excepción" respectivamente, cuyo formato es idéntico al presentado en el cuadro 6-5:

Cuadro 6-5/X.781 – Formulario de soporte de tipo de datos

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional

donde:

- El campo "Índice" es la referencia a cada identificador de campo de operación que aparece en el cuadro 6-1.
- El campo "Subíndice" es la referencia al campo vástago contenido en el tipo de datos super.
- El campo "Identificador de campo" describe cada nombre de campo vástago de un tipo de datos construido.
- El campo "Constricciones y valores" es la descripción del tipo de datos de cada campo vástago. Si el tipo del campo es complejo, habrá más comentarios para mostrar sus campos vástagos anidados en la fila siguiente a la que ocupa, y este proceso continuará hasta que los tipos de campo resulten ser tipos básicos predefinidos en una especificación CORBA.
- El campo "Información adicional" sirve para especificar alguna información precisa que deba ser declarada por el implementador. Cuando el tipo de un campo es "any", se debe insertar un número de índice de referencia en este campo. Cuando el campo es un miembro de estado de un valuetype, se describirá en este campo su publicidad y herencia. Véase 7.5 e) para más detalles.

7 Instrucciones para rellenar el formulario CIICS

En esta cláusula se presentan las instrucciones para rellenar cada columna definida en 6.2 y sus subcláusulas.

7.1 Definición de soportado

Se dice que una capacidad está soportada si la implementación sometida a prueba puede realizar la funcionalidad especificada.

7.2 La columna Status

Esta columna indica el nivel de soporte requerido para la conformidad con una determinada especificación IDL. Los valores son los definidos en 6.2.1.

Directrices para rellenar este campo:

- a) Para el "Cuadro de soporte de interfaces", si es obligatorio, "Status" se debe rellenar con 'm'; si es facultativo, "Status" se debe rellenar con 'o'. Si una interfaz no es ejemplificable, "Status" se debe rellenar con "-", lo que significa que no es aplicable.
- b) Para el "Cuadro de soporte de atributo", se debe insertar 'm' en la "columna status" para la totalidad de "Obtener" ("Get") , y también para "Fijar" ("Set") si el atributo es normal; se debe insertar '-' en este campo para "Fijar" si el atributo viene precedido de la palabra clave "readonly" (lectura solamente).

- c) Para el "Cuadro de soporte de operación", se debe insertar 'm' en este campo si la operación es necesaria para la funcionalidad de gestión; de lo contrario, se podría insertar 'o'. Si la operación es obligatoria, para cada "campo de operación" de esta operación, la columna "status" se debe rellenar con 'm'; si no lo es, debe insertarse 'c:m'.
- d) Para el "Cuadro de soporte de tipo de datos", sólo cuando un tipo sea el tipo "union" (unión), la columna "status" de su campo vástago podría rellenarse con 'o', puesto que las uniones son utilizadas principalmente para indicar opciones, en especial cuando existe una sola rama con "VERDADERO" ("TRUE") como valor de discriminación. Para todos los demás tipos, se debe insertar 'm' en esta columna. Por supuesto, cuando un ítem que contiene otros tipos es facultativo, todos los ítems contenidos por aquél debe añadir "c:" como prefijo para este campo, como se expresa en 6.2.1.

7.3 La columna Soporte

Esta columna debe ser rellenada por el suministrador o implementador para indicar el nivel de implementación de cada ítem. En 6.2.1 se indican las notaciones disponibles para este campo.

Las siguientes son directrices para rellenar este campo:

- a) Si se pretende que un ítem está "soportado", todos los ítems obligatorios que aquél contiene también deben estar soportados. Si no, la columna "Soporte" puede rellenarse con 'N'.
- b) Si la columna "Status" de un ítem está rellenada con '-', la única notación para la columna Soporte correspondiente es '-'.
- c) En los cuadros de formulario CIICS, cada ítem marcado con 'm' debe estar soportado por la IUT.

7.4 La columna "Constricciones y valores"

Las columnas "Constricciones y valores" de los cuadros (que deben ser rellenadas en la especificación de formulario cuando ello sea aplicable) contienen las constricciones y valores del ítem específico (por ejemplo, atributo, campo de operación). Esta información puede incluir, **cuando sea aplicable**:

- a) toda constricción relacionada con el soporte del ítem en particular;
- b) valores específicos para los parámetros de atributos u operaciones que están soportados;
- c) los tipos permitidos de acuerdo con la especificación de la norma.

Cuando el ítem indique un tipo, el nombre de tipo de este ítem debe ser insertado en esta columna. Esta regla es adecuada para el "Cuadro de soporte de atributo", "Cuadro de soporte de operación" y "Cuadro de soporte de tipo de datos".

El nombre de tipo utilizado para rellenar la columna puede describirse mediante la siguiente expresión de generación:

<type name> ::= <basic type name> | <user-defined type name>

<basic type name> ::= **void | short | unsigned short | long | unsigned long | long long | unsigned long long | float | double | long double | char | boolean | octet | string | TypeCode | objref | any | wchar | wstring | fixed**

<user-defined type name> ::= cadena que denota un nombre de tipo absoluto

Para cada tipo de datos definido por el usuario se sugiere también insertar en esta columna la "type kind", encerrada entre paréntesis.

Type kind es la descripción abstracta de los tipos IDL. La kind de cada tipo básico tiene el mismo nombre que su nombre de tipo. Los nombres de tipo de todos los tipos definidos por el usuario son: "struct", "union", "sequence", "array", "enum", "interface", "valuetype", "exception", "bstring"².

7.5 La columna "Información adicional"

Esta columna debe ser rellena por el suministrador o implementador con el fin de explicar alguna información específica acerca de la implementación que no esté incluida en otras columnas. Esta Recomendación presenta algunas reglas generales que deben ser observadas por los suministradores, a saber:

- a) En el "Cuadro de soporte de interfaz", se debe especificar explícitamente en este campo el papel de servidor o cliente. Cuando se actúa como servidor, se debe insertar "As server" en este campo; si no, se debe insertar "As client" en este campo. Cuando el papel desempeñado por una interfaz está determinado, todas las operaciones y atributos contenidos en esta interfaz desempeñarán el mismo papel.
- b) En el "Cuadro de soporte de atributo", si un atributo tiene valores por defecto cuando el objeto es creado, se sugiere añadir el valor en este campo. Si un atributo es readonly (de lectura solamente), y su valor inicial se da en el momento de la creación del objeto, se debe insertar "set by create" en este campo. Si un atributo ha sido heredado de otra interfaz, se insertará "Inherited" en este campo.
- c) Para el "Cuadro de soporte de operación", si una operación ha sido heredada de otra interfaz, se debe insertar "Inherited" en este campo. Si algún parámetro tiene valores por defecto, esto debe ser indicado en la fila correspondiente a este campo para el parámetro. Por cada excepción definida por el usuario, la condición cuyo cumplimiento provoca la excepción debe ser indicada en este campo.
- d) En el "Cuadro de soporte de atributo" y "Cuadro de soporte de operación", si un atributo u operación está sobrecargado en una interfaz vástago, se debe indicar "overload" en este campo.
- e) En el "Cuadro de soporte de atributo", "Cuadro de soporte de operación" y "Cuadro de soporte de tipo de datos", cuando el tipo de un campo es "any", se debe incluir en este campo una referencia, que indique el número de índice de un cuadro de apéndice, en el cual deberán estar completamente descritos los posibles tipos en fase de ejecución que habrán de ser insertados en este "any". El número de índice de apéndice recomendado es similar al índice en la primera columna de la misma fila, excepto que se le añade un carácter "A" como prefijo. El "Cuadro de soporte de apéndice de tipo any" tiene el mismo formato del "Cuadro de soporte de tipo de datos" (para más detalles, véanse los ejemplos presentados en el anexo C).
- f) En el "Cuadro de soporte de tipo de datos", cuando el campo es un miembro de estado de un valuetype, la publicidad y herencia de este miembro de estado deben estar descritas. La publicidad puede consistir en "estado público" ("public state") o "estado privado" ("private state"); y si este miembro de estado ha sido heredado de su tipo de base, deberá existir en este campo la expresión "inherited from <base valuetype name>". En el anexo D se presenta un ejemplo de un formulario CIICS para los valuetypes.

² "bstring" significa "bounded string" (cadena acotada). En IDL, "bounded string" especifica una cadena limitada en su longitud. En CIICS, la bstring <n> se utiliza para describir este tipo, donde "n" es la longitud de la cadena (string).

- g) El tipo de Excepciones de Sistema que utiliza cada operación, si las utiliza, debe ser explicado en este campo para la operación. Si la Excepción de Sistema NO_IMPLEMENT puede ser utilizada para especificar "not supported", el "Status" correspondiente debe ser rellenado con 'o'.
- h) Para cada cuadro CIICS, si el suministrador tiene algo especial que reivindicar, se podría utilizar este campo. Si el espacio no es suficiente, se deben agregar cuadros ampliados como parte del formulario de CIICS.

7.6 La columna "Índice"

Cada línea del formulario CIICS está numerada en su borde izquierdo. Esta numeración se incluye como un medio de identificación unívoca de todos los detalles de implementación posibles dentro del formulario CIICS.

La manera de referirse a una respuesta individual viene dada por la secuencia siguiente:

- a) una referencia a la supercláusula del ítem;
- b) el carácter de separación ".";
- c) un número único.

En el anexo A se presenta un ejemplo de la utilización de esta notación.

La columna "índice" exhibe la relación de contención entre las estructuras de sintaxis IDL. La relación de contención se ilustra en 6.1 "Visión general del IDL".

7.7 La columna de "Subíndice"

Esta columna tiene el mismo significado y formato que la columna "Índice". De igual manera, está constituida de números consecutivos:

- a) En el "Cuadro soporte de operación", la supercláusula de "Subíndice" es la columna "Índice" que aparece antepuesta en la misma fila.
- b) En el "Cuadro soporte de tipo de datos", la supercláusula de "Subíndice" es el índice de su ítem contenedor directo, que puede estar tanto en la columna "Índice" como en la columna "Subíndice".

7.8 La columna "Categoría"

Esta columna del "formulario Soporte de operación" puede rellenarse eligiendo entre "parámetro" y "terminación". El "parámetro" indica que el "campo operación" es un parámetro de invocación de operación (que incluye el contexto), mientras que la "terminación" indica que el "campo operación" es un tipo de situaciones de terminación.

7.9 La columna "Propiedad de campo"

Esta columna corresponde a la columna "Categoría". Si el campo "Categoría" es "parámetro" esta columna podría ser "in", "out", "inout" o "context", que indican el modo del parámetro. Si el campo "Categoría" es "terminación", esta columna podría ser "success reply" o "exception". "Success reply" es la respuesta deseada de la operación, que puede ser un valor de retorno o "void"; "exception" indica que la operación ha abortado por alguna razón particular.

7.10 Las columnas "Identificador de interfaz", "Identificador de atributo", "Identificador de operación"

Estas columnas deben ser rellenas con los nombres de interfaces, atributos u operaciones correspondientes.

NOTA – "Identificador de interfaz" es el nombre absoluto de la interfaz, que puede ser identificado unívocamente dentro del espacio de nombre del modelo de información IDL completo. "Identificador de atributo" e "Identificador de operación", sólo son nombres relativos del ítem cuyo espacio de nombre está justamente en la interfaz contenedora.

7.11 La columna "Interfaz derivada"

La especificación IDL soporta las relaciones de herencia. Esta columna deja el espacio que ocuparían las interfaces derivadas, si las hubiera.

NOTA – Solamente las interfaces progenitoras directas deben ser rellenadas en esta columna; las interfaces ancestrales no es necesario insertarlas en este campo.

ANEXO A

Ejemplo de especificación de formulario CIICS

A.1 Definición en CORBA IDL

En el presente anexo se ilustran los formularios CIICS para "Soporte de interfaz", "Soporte de atributo" y "Soporte de operación". Sigue un ejemplo sencillo de una definición IDL:

```
// DESCRIPTION:
//
// source IDL file for simple BANK example
//
//*****
module BankModule {
//
// a simple description of a bank account
//
    interface account {
        readonly attribute float balance;

        void makeLodgement (in float f);
        void makeWithdrawal (in float f);
    };
//
// a simple description of a bank current account
//
    interface currentAccount : account {
        readonly attribute float overdraftLimit;
    };
//
// a bank simply manufactures accounts
//
// bank::reject is raised if a duplicate account name is seen
//
    interface bank {
        exception reject {};
        account newAccount (in string name) raises (reject);
        currentAccount newCurrentAccount (in string name, in float limit)
            raises (reject);
        void deleteAccount (in account a);
    };
};
```

A.2 CIICS

De acuerdo con la anterior definición IDL, la CIICS para este simple modelo de información IDL puede ilustrarse mediante los cuadros A.1 a A.3.

Cuadro A.1/X.781 – Interface BankModule::bank support

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
1	BankModule::bank		m	Y	Como servidor

Cuadro A.1.1/X.781 – BankModule::bank attribute support

No hay definición de atributo en interfaz de banco.

Cuadro A.1.2/X.781 – BankModule::bank operation support (como servidor)

Índice	Identificador de operación	Status	Soporte	Información adicional
1.2.1	newAccount	m	Y	
1.2.2	newCurrentAccount	m	Y	
1.2.3	deleteAccount	m	Y	

Índice	Subíndice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
1.2.1	1.2.1.1	name	String	parameter	in	m	Y	
	1.2.1.2		BankModule::account (interface)	termination	success reply	m	Y	
	1.2.1.3	reason		termination	exception	m	Y	Cuando ha existido el objeto account con el mismo nombre
1.2.2	1.2.2.1	name	String	parameter	in	m	Y	
	1.2.2.2	limit	Float	parameter	in	m	Y	
	1.2.2.3		BankModule::currentAccount (interface)	termination	success reply	m	Y	
	1.2.2.4	reason		termination	exception	m	Y	Cuando ha existido el objeto current-Account con el mismo nombre
1.2.3	1.2.3.1	a	BankModule::account (interface)	parameter	in	m	Y	
	1.2.3.2		Void	termination	success reply	m	Y	

NOTA – Si el parámetro de invocación tiene valor por defecto, esto se indica en la columna "Constricciones y valores".

Cuadro A.2/X.781 – Interface BankModule::account support

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
2	BankModule::account		m	Y	Como servidor

Cuadro A.2.1/X.781 – BankModule::account attribute support

Índice	Identificador de atributo	Constricciones y valores	Obtener		Fijar		Información adicional
			Status	Soporte	Status	Soporte	
2.1.1	balance	float	m	Y	-	-	El valor por defecto es 0 cuando se crea el objeto

NOTA – Puesto que el atributo "balance" es un atributo "readonly", sólo el "Get Status" es obligatorio, y el "Set Status" no es aplicable.

Cuadro A.2.2/X.781 – BankModule::account operation support (como servidor)

Índice	Identificador de operación	Status	Soporte	Información adicional
2.2.1	makeLodgement	m	Y	
2.2.2	makeWithdrawal	m	Y	

Índice	Subíndice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
2.2.1	2.2.1.1	f	float	parameter	in	m	Y	
	2.2.1.2		void	termination	success reply	m	Y	
2.2.2	2.2.2.1	f	float	parameter	in	m	Y	
	2.2.2.2		void	terminator	success reply	m	Y	

Cuadro A.3/X.781 – Interface BankModule::currentAccount support

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
3	BankModule::current Account	BankModule::account	m	Y	Como servidor

Cuadro A.3.1/X.781 – BankModule::currentAccount attribute support

Índice	Identificador de atributo	Constricciones y valores	Obtener		Fijar		Información adicional
			Status	Soporte	Status	Soporte	
3.1.1	balance	float	m	Y	-	-	Heredado
3.1.2	overdraft Limit	float	m	Y	-	-	El valor es fijado por create

NOTA – Puesto que el atributo "balance" es heredado de la interfaz BankModule::account, "Inherited" se especifica en la "Información adicional".

Si un atributo u operación está sobrecargado en una interfaz vástago, se debe indicar "overload" en el campo "Información adicional".

Cuadro A.3.2/X.781 – BankModule::currentAccount operation support (como servidor)

Índice	Identificador de operación	Status	Soporte	Información adicional
3.2.1	makeLogement	m	Y	Heredado
3.2.2	makeWithdrawal	m	Y	Heredado

Índice	Sub-índice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
3.2.1	3.2.1.1	f	float	parameter	in	m	Y	
	3.2.1.2		void	termination	success reply	m	Y	
3.2.2	3.2.2.1	f	float	parameter	in	m	Y	
	3.2.2.2		void	termination	success reply	m	Y	

El anterior es un ejemplo muy simple que muestra la especificación de formulario CIICS para "Formulario de soporte de interfaz", "Formulario de soporte de atributo" y "Formulario de soporte de operación". En el anexo B se muestra un ejemplo que ilustra el "Formulario de soporte de tipo de datos".

ANEXO B

Ejemplo de una especificación de "Formulario de soporte de tipo de datos"

B.1 Definición IDL CORBA

En este anexo se presenta un ejemplo de IDL que contiene varios tipos de datos complejos definidos por el usuario en IDL, que puede utilizarse para mostrar el formato de "Formulario de soporte de tipo de datos". La siguiente es la definición IDL:

```

module typeExample {

    typedef short shortArray[2][3];
    typedef sequence<short,4> shortSequence;

    enum Color { red, blue, green};
    exception ExceptionType {
        short    number;
        string   reason;
    };

    typedef short short_alias;

    struct DataStructure {
        short_alias  s;
        float        f;
    };

    union UN_DS {
        short s;
        DataStructure ds;
    };

    interface typeInterface {
        UN_DS op (
            in      shortArray      p1_sA,
            in      shortSequence    p2_sS,
            inout   Color            p3_color,
            out     DataStructure     p4_UT
        )
        raises (ExceptionType)
        context ( "key1", "key2" ) ;
    };
};

```

B.2 CIICS

De acuerdo con la anterior definición IDL, se puede ilustrar el modelo de información de este ejemplo mediante los cuadros siguientes:

Cuadro B.1/X.781 – typeExample::typeInterface interface support

Índice	Indicador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
1	TypeExample::typeInterface		m	Y	Como servidor

Cuadro B.1.1/X.781 – typeExample::typeInterface attribute support

La interfaz typeExample::typeInterface no tiene atributo definido.

Cuadro B.1.2/X.781 – typeExample::typeInterface operation support

Índice	Identificador de operación	Status	Soporte	Información adicional
1.2.1	Op	m	Y	

Índice	Sub-índice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
1.2.1	1.2.1.1	p1_sA	::typeExample::shortArray (array[2][3])	parameter	in	m	Y	
	1.2.1.2	p2_sS	::typeExample::shortSequence (sequence<4>)	parameter	in	m	Y	
	1.2.1.3	p3_color	::typeExample::Color (enum)	parameter	inout	m	Y	
	1.2.1.4	p4_UT	::typeExample::DataStructure (struct)	parameter	out	m	Y	
	1.2.1.5		key1	parameter	context	m	Y	
	1.2.1.6		key2	parameter	context	m	Y	
	1.2.1.7		::typeExample::UN_DS (union)	termination	success reply	m	Y	
	1.2.1.8	Exception Type	::typeExample::ExceptionType (exception)	termination	exception	m	Y

Subcuadro de soporte de parámetro

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1.2.1.1	1.2.1.1.1		Short	m	Y	
1.2.1.2	1.2.1.2.1		Short	m	Y	
1.2.1.4	1.2.1.4.1	s	::typeExample::short_alias (short)	m	Y	
	1.2.1.4.2	f	Float	m	Y	

NOTA – El campo vástago de un tipo union debe contener al menos un ítem como su campo vástago obligatorio. Aquí se supone que el campo "s" es obligatorio, y que "ds" es facultativo. De hecho, que un campo vástago de tipo "union" sea obligatorio o facultativo depende de la semántica de la implementación.

Subcuadro de soporte de valor de retorno

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1.2.1.7	1.2.1.7.1	S	Short	m	Y	
	1.2.1.7.2	Ds	::typeExample::DataStructure (struct)	o	Y	
	1.2.1.7.2.1	S	::typeExample::short_alias (short)	c:m	Y	
	1.2.1.7.2.2	F	Float	c:m	Y	

NOTA – El campo vástago de un tipo union contendrá al menos un ítem como su campo vástago obligatorio. De hecho, que un campo vástago de tipo "union" sea obligatorio o facultativo depende de la semántica de la implementación. Aquí se supone que el campo "f" es obligatorio y que el campo "ds" es facultativo. Por tanto, cada "Status" de los ítems contenidos en "ds" debe ir precedido por "c:", lo que significa que solamente cuando esté soportado "ds" podrían ser obligatorios los campos vástagos "s" y "f".

Subcuadro de soporte de excepción

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1.2.1.8	1.2.1.8.1	number	short	m	Y	
	1.2.1.8.2	reason	string	m	Y	

ANEXO C

Ejemplo que muestra la especificación del formulario "Appendix any type support proforma"

C.1 Definición CORBA IDL

En este anexo se presenta un ejemplo de IDL que contiene una interfaz que tiene un atributo de tipo "any" y una operación con un parámetro any-typed, que podría ser utilizado para presentar el formato de "Appendix any type support proforma". He aquí la definición IDL:

```
Module anyExample {
    struct DataStructure {
        short    s;
        string   str;
    };

    interface anyInterface {
        attribute any anyAttr;
        void op ( inout any pl ) ;
    };
};
```

C.2 CIICS

De acuerdo con la anterior definición IDL, la CIICS del ejemplo de modelo de información se puede ilustrar en los cuadros siguientes:

Cuadro C.1/X.781 – anyExample::anyInterface interface support

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
1	anyExample::anyInterface		m	Y	Como servidor

Cuadro C.1.1/X.781 – anyExample ::anyInterface attribute support

Índice	Identificador de atributo	Constricciones y valores	Obtener		Fijar		Información adicional
			Status	Soporte	Status	Soporte	
1.1.1	anyAttr	any	m	Y	m	Y	Véase apéndice Cuadro C.1.1.1

NOTA – Puesto que el tipo del atributo "anyAttr" es "any", el campo "Información adicional" se rellena con el número de índice de referencia de apéndice del cuadro de apéndice correspondiente: "C.1.1.1", que se compone de una "A" seguida del índice de este atributo "1.1.1". Aquí se supone que los posibles tipos permitidos en la fase de ejecución (run-time), para este atributo son "unsigned short", "long" y "float".

Cuadro C.1.2/X.781 – anyExample::anyInterface operation support

Índice	Identificador de operación	Status	Soporte	Información adicional
1.2.1	op	m	Y	Como servidor

Índice	Sub-índice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
1.2.1	1.2.1.1	p1	any	parameter	inout	m	Y	Véase apéndice Cuadro C.1.2.1.1
	1.2.1.2		void	termination	Success reply	m	Y	

NOTA – Puesto que el parámetro "p1" es de tipo "any", el campo "Información adicional" de este ítem se rellena con el número de índice de referencia de apéndice del cuadro de apéndice correspondiente: "C.1.2.1.1", que se compone de una "A" seguida del subíndice de este parámetro "1.2.1.1". Aquí se supone que los posibles tipos permitidos en la fase de ejecución (run-time) para este parámetro son "string" y "anyExample::DataStructure".

A continuación se presentan los cuadros de apéndice para los tipos any antes descritos.

Cuadro C.1.2.1/X.781 – <::anyExample::anyInterface::anyAttr → anyAttr any types support>

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1	1.1		long	m	Y	
2	2.1		ushort	m	Y	
3	3.1		float	m	Y	

Cuadro C.1.2.1.1/X.781 – <::anyExample::anyInterface::op → p1 any types support>

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1	1.1		string	m	Y	
2	2.1		anyExample::dataStructure (struct)	m	Y	
	2.1.1	s	short	m	Y	
	2.1.2	str	string	m	Y	

NOTA – El "Cuadro de soporte de tipo any de apéndice" tiene el mismo formato del "Cuadro de soporte de tipo datos", y contiene todos los tipos permitidos que pueden insertarse en este campo.

ANEXO D

Ejemplo que muestra la especificación de "formulario de soporte valuetype"

D.1 Definición CORBA IDL

En este anexo se presenta un ejemplo de IDL que contiene la definición valuetype con herencia, y está definido como un parámetro de una operación de otra interfaz, que podría ser utilizada para describir el "formulario de soporte valuetype". Ésta es la definición IDL:

```

Module valueModule {
    valuetype BaseValue {
        public short id;
    };
    valuetype ValueExample : BaseValue {
        private string name;
        public long value;

        factory init(in string name, in long value);
        //method
        void local_op ();
    };

    interface aInterface {
        void remote_op (in ValueExample p1);
    };
};

```

D.2 CIICS

De acuerdo con la anterior definición IDL, la CIICS para este ejemplo de modelo de información puede ilustrarse en los cuadros siguientes:

Cuadro D.1/X.781 – valueModule::aInterface interface support

Índice	Identificador de interfaz	Interfaz derivada	Status	Soporte	Información adicional
1	valueModule::aInterface		m	Y	Como servidor

Cuadro D.1.1/X.781 – valueModule::aInterface attribute support

.La interfaz valueModule::ValueInterface no tiene atributo definido.

Cuadro D.1.2/X.781 – valueModule::aInterface operation support

Índice	Identificador de operación	Status	Soporte	Información adicional
1.2.1	remote_op	m	Y	Como servidor

Índice	Sub-índice	Identificador de campo de operación	Constricciones y valores	Categoría	Propiedad de campo	Status	Soporte	Información adicional
1.2.1	1.2.1.1	p1	ValueModule::ValueExample (valuetype)	parameter	in	m	Y	
	1.2.1.2		void	termination	Success reply	m	Y	

Subcuadro de soporte de parámetro

Índice	Subíndice	Identificador de campo	Constricciones y valores	Status	Soporte	Información adicional
1.2.1.1	1.2.1.1.1	Id	short	m	Y	Public state, inherited from valueModule::BaseValue
1.2.1.2	1.2.1.2.1	Name	string	m	Y	Private state
	1.2.1.2.2	Value	long	m	Y	Public state

NOTA – Un examen de los anteriores cuadros muestra que, cuando se transfiere como parámetro, valuetype se trata de una manera muy similar a aquella en que se trata struct, excepto que valuetype puede contener estados heredados de sus valuetypes de base. Puesto que "id" es un miembro de estado público que valueModule::ValueExample heredó de valueModule::BaseValue, el "estado público, heredado de valueModule::BaseValue" se inserta en el campo "Información adicional". Asimismo, puesto que "name" es un miembro de estado privado ordinario de valueModule::ValueExample, el "estado privado" se inserta en el campo "Información adicional".

Aunque hay operaciones de inicializador "init()" y operaciones ordinarias local_op() definidas en este valuetype valueModule::ValueExample, dichas operaciones no aparecen en el formulario ICS porque todas ellas son operaciones locales que están fuera del alcance de las declaraciones de conformidad de implementación de interfaz.

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 multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
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 y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación