



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# UIT-T

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

# Z.100

**Corrigendum 1**  
(08/2004)

SERIE Z: LENGUAJES Y ASPECTOS GENERALES  
DE SOPORTE LÓGICO PARA SISTEMAS DE  
TELECOMUNICACIÓN

Técnicas de descripción formal – Lenguaje de  
especificación y descripción

---

Lenguaje de especificación y descripción

**Corrigendum 1**

Recomendación UIT-T Z.100 (2002) – Corrigendum 1

---

RECOMENDACIONES UIT-T DE LA SERIE Z

**LENGUAJES Y ASPECTOS GENERALES DE SOPORTE LÓGICO PARA SISTEMAS DE  
TELECOMUNICACIÓN**

<b>TÉCNICAS DE DESCRIPCIÓN FORMAL</b>	
<b>Lenguaje de especificación y descripción</b>	<b>Z.100–Z.109</b>
Aplicación de técnicas de descripción formal	Z.110–Z.119
Gráficos de secuencias de mensajes	Z.120–Z.129
Lenguaje ampliado de definición de objetos	Z.130–Z.139
Notación de prueba y de control de prueba	Z.140–Z.149
Notación de requisitos de usuarios	Z.150–Z.159
<b>LENGUAJES DE PROGRAMACIÓN</b>	
CHILL: el lenguaje de alto nivel del UIT-T	Z.200–Z.209
<b>LENGUAJE HOMBRE-MÁQUINA</b>	
Principios generales	Z.300–Z.309
Sintaxis básica y procedimientos de diálogo	Z.310–Z.319
LHM ampliado para terminales con pantalla de visualización	Z.320–Z.329
Especificación de la interfaz hombre-máquina	Z.330–Z.349
Interfaces hombre-máquina orientadas a datos	Z.350–Z.359
Interfaces hombre-máquina para la gestión de las redes de telecomunicaciones	Z.360–Z.369
<b>CALIDAD</b>	
Calidad de soportes lógicos de telecomunicaciones	Z.400–Z.409
Aspectos de la calidad de las Recomendaciones relativas a los protocolos	Z.450–Z.459
<b>MÉTODOS</b>	
Métodos para validación y pruebas	Z.500–Z.519
<b>SOPORTE INTERMEDIO</b>	
Entorno del procesamiento distribuido	Z.600–Z.609

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

# Recomendación UIT-T Z.100

## Lenguaje de especificación y descripción

### Corrigendum 1

#### Resumen

En el presente corrigendum se introducen varias modificaciones a la Rec. UIT-T Z.100 (08/2002) y a la enmienda 1 a la Z.100 (10/2003) para corregir y aclarar algunos aspectos de esa Rec. UIT-T Z.100 y de su anexo B.

Las modificaciones se dividen en tres partes:

- modificaciones al texto principal (y al anexo D) de la Rec. UIT-T Z.100 publicada como Z.100 (08/2002);
- sustitución del anexo B.9;
- adiciones al anexo B.

En cada modificación del texto principal (y del anexo D) de la Z.100 se indica si se trata de una aclaración, corrección de alguna deficiencia, ampliación o modificación (véase el apéndice II/Z.100) mediante un título seguido del número de cláusula de la Z.100 donde se encuentra el texto que se ha de modificar. En muchos casos también se incluye la subdivisión (*Gramática abstracta, gramática concreta, semántica o modelo*) del texto de la Z.100. El título de cada modificación concluye con una breve descripción de la misma.

Los cambios que hay que introducir al anexo B son necesarios para añadir la sintaxis tradicional de las definiciones de tipos de datos, que fueron omitidas en la primera publicación de este anexo en la enmienda 1 a la Z.100 (10/2003).

#### Orígenes

El corrigendum 1 a la Recomendación UIT-T Z.100 (2002) fue aprobado el 29 de agosto de 2004 por la Comisión de Estudio 17 (2001-2004) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

## 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.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

## 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 2005

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

# ÍNDICE

	Página
1 Modificaciones del texto principal de la Rec.UIT-T Z.100 (y del anexo D).....	1
1.1 Ampliación – 6.6 – Gramática concreta – Se permite que <page number area> esté vacío .....	1
1.2 Corrección de deficiencia – 7.2 – Gramática concreta – Se sustituye conected por attached .....	1
1.3 Corrección de deficiencias – 8.1.1.1 – Gramática concreta – Sintaxis incorrecta .....	1
1.4 Aclaración – 8.1.1.1 – Modelo – Virtualidad de los estados compuestos implícitos .....	1
1.5 Corrección de deficiencia – 8.1.1.5 – Gramática concreta – Sintaxis incorrecta .....	2
1.6 Corrección de deficiencias – 8.1.1.5 – Gramática concreta – Sintaxis incorrecta .....	2
1.7 Corrección de deficiencias – 8.1.5 – Gramática concreta – Restricción innecesaria .....	2
1.8 Corrección de deficiencia – 8.3.1 – Gramática concreta – Sustitución de "connected" por "attached" .....	2
1.9 Corrección de deficiencias – 8.5 – Gramática concreta – Sustitución de "connected" por "attached" .....	2
1.10 Corrección de deficiencias – 9 – Gramática concreta – Sintaxis incorrecta ..	2
1.11 Corrección de deficiencia – 9 – Gramática concreta – Sustitución de "connected" por "attached" .....	3
1.12 Aclaración/corrección de deficiencia – 9 – Semántica – Máquina de estado predeterminada .....	3
1.13 Corrección de deficiencias – 9.2 – Gramática concreta – Sintaxis incorrecta .....	3
1.14 Corrección de deficiencias – 9.3 – Gramática concreta – Sintaxis incorrecta .....	3
1.15 Aclaración – 9.5 – Gramática concreta – Sintaxis incorrecta .....	4
1.16 Corrección de deficiencias – 10.1, 10.2 – Gramática concreta – Sustitución de "connected" por "attached" .....	4
1.17 Aclaración – 11.2 – Gramática concreta – Sintaxis incorrecta .....	4
1.18 Aclaración – 11.2 – Gramática concreta – Utilización de <composite state name> .....	5
1.19 Aclaración – 11.2 – Modelo – Utilización de <composite state name> .....	5
1.20 Corrección de deficiencias – 11.3 – Gramática concreta – Sintaxis incorrecta .....	5
1.21 Corrección de deficiencias – 11.6 – Gramática concreta – Sintaxis incorrecta .....	5
1.22 Corrección de deficiencias – 11.9 – Gramática concreta – Sintaxis incorrecta .....	6
1.23 Aclaración – 11.11 – Utilización de <composite state name> .....	6

1.24	Aclaración – 11.11 – Gramática abstracta – Procedimiento con estados para procedimientos de entrada/salida.....	6
1.25	Corrección de deficiencias – 11.11.1 – Gramática concreta – Sintaxis incorrecta .....	6
1.26	Aclaración – 11.11.1 – Gramática concreta – Utilización de < <u>composite state name</u> > .....	6
1.27	Aclaración – 11.11.1 – Gramática concreta – Sintaxis incorrecta .....	7
1.28	Corrección de deficiencias – 11.11.2 – Gramática concreta – Sintaxis incorrecta .....	7
1.29	Aclaración – 11.11.2 – Gramática concreta – Utilización de < <u>composite state name</u> > .....	7
1.30	Corrección de deficiencias – 11.11.2 – Gramática concreta – Permite puertas en la partición de estado heredada .....	7
1.31	Corrección de deficiencia – 11.11.2 – Gramática concreta – Sintaxis incorrecta .....	7
1.32	Corrección de deficiencias – 11.11.3 – Gramática concreta – Sintaxis incorrecta .....	8
1.33	Aclaración – 11.11.4 – Gramática concreta – Sintaxis incorrecta .....	8
1.34	Aclaración – 11.12.2.2 – Gramática concreta – Sintaxis incorrecta .....	8
1.35	Aclaración – 11.13.3 – Gramática concreta – Ambigüedad.....	8
1.36	Modificación – 11.13.4 – Semántica – Encaminamiento de señalización .....	8
1.37	Modificación – 12.1.2 – Modelo de interfaces implícitas .....	9
1.38	Corrección de deficiencias – 12.1.8 – Gramática concreta – Sintaxis incorrecta .....	10
1.39	Aclaración – 12.3.4.1 – Semántica – Unidad de tiempo .....	10
1.40	Aclaración – 13.1 – Gramática concreta – Sintaxis perfeccionada .....	10
1.41	Aclaración – D.3.11.2, D.3.12.2 – Unidad de tiempo .....	10
2	Modificación del B.9 (anexo B, "Compatibilidad con sistemas anteriores") .....	11
B.9	Comportamiento de las operaciones.....	11
3	Adiciones al anexo B – Compatibilidad con sistemas anteriores .....	11
B.11	Definición de datos.....	12
B.12	Definición de tipo de datos.....	12
B.13	Syntypes .....	13

# Recomendación UIT-T Z.100

## Lenguaje de especificación y descripción

### Corrigendum 1

#### 1 Modificaciones del texto principal de la Rec. UIT-T Z.100 (y del anexo D)

Cada uno de los títulos de esta cláusula consta de:

- el número de título, por ejemplo 1.1;
- una indicación del tipo de modificación de que se trata, es decir, aclaración, corrección de deficiencia, ampliación o modificación;
- el número de cláusula que se va a modificar;
- como opción, la subdivisión de texto (*Gramática abstracta, gramática concreta, semántica o modelo*);
- una breve descripción de la modificación.

##### 1.1 Ampliación – 6.6 – Gramática concreta – Se permite que <page number area> esté vacío

En la gramática concreta insértese "[" justo antes de "<implicit text symbol>" y *suprímase* "[" antes de "<page number>" para formar la regla:

[ <implicit text symbol> *contains* <page number> [ (<number of pages>) ] ]

##### 1.2 Corrección de deficiencia – 7.2 – Gramática concreta – Se sustituye "conected" por "attached"

En la regla <package dependency area> sustitúyase "*connected*" por "*attached*" porque el diagrama de lote o el área de referencia lo produce la regla del diagrama que lo encierra.

##### 1.3 Corrección de deficiencias – 8.1.1.1 – Gramática concreta – Sintaxis incorrecta

En la regla <agent type diagram> *insértese* "{ " antes de "<system type diagram>", *insértese* "}" después de <process type diagram> e *insértese* "]" después de <package use area> para formar la regla:

<agent type diagram> ::=  
{ <system type diagram> | <block type diagram> | <process type diagram> }  
[ *is associated with* <package use area> ]

##### 1.4 Aclaración – 8.1.1.1 – Modelo – Virtualidad de los estados compuestos implícitos

Para aclarar que una máquina de estados implícitos de un tipo de agente puede redefinirse si el tipo de agente puede redefinirse, se introducen los siguientes cambios en el texto.

###### En 8.1.1.1 Modelo

Al final del párrafo que termina por " se representa por la <agent body area>." *insértese* la siguiente frase "La virtualidad de la definición del estado compuesto es la misma que la virtualidad del tipo de agente".

En el párrafo siguiente al anterior, después de los guiones, *sustitúyase*: "tipo de estado compuesto implicado virtual" *por* "tipo de estado compuesto implicado con la misma virtualidad que el tipo de agente".

## 1.5 Corrección de deficiencia – 8.1.1.5 – Gramática concreta – Sintaxis incorrecta

En la regla <composite state type diagram> *insértese* el " }" que falta después del elemento <composite state structure area>.

## 1.6 Corrección de deficiencias – 8.1.1.5 – Gramática concreta – Sintaxis incorrecta

En la regla <composite state type diagram> *sustitúyase* "**associated with**" antes de { <state connection point>\* } *set* por "**connected to**" porque los elementos están unidos (como puertas) y además el texto los describe como que están "conectados". Además, *sustitúyase cada vez que aparece* <state connection point> en la Z.100 por <state connection point area> para conservar la coherencia con la metagramática gráfica.

## 1.7 Corrección de deficiencias – 8.1.5 – Gramática concreta – Restricción innecesaria

*Suprímase* el último párrafo de la cláusula justo antes de la semántica, cuyo texto es:

El <interface identifier> de una <interface gate definition> no deberá identificar la interfaz definida implícitamente por la entidad a que está conectada la puerta (véase 12.1.2).

## 1.8 Corrección de deficiencia – 8.3.1 – Gramática concreta – Sustitución de "connected" por "attached"

En la regla <specialization area> *sustitúyase* "connected" por "attached" porque la zona de referencia del tipo la produce la regla del diagrama que la encierra.

## 1.9 Corrección de deficiencias – 8.5 – Gramática concreta – Sustitución de "connected" por "attached"

La <association area> ha de estar "attached" (vinculada) a las <linked type reference area> porque la zona de referencia la genera la regla del diagrama que la encierra.

*Sustitúyase* la regla <association area> por

```
<association area> ::=
    <association symbol>
    [ is associated with <association name> ]
    is attached to { <linked type reference area> <linked type reference area> } set
    is associated with { <association end area> <association end area> } set
```

y *añádase* el siguiente párrafo después de esta regla:

"Las <linked type reference area> están vinculadas a un extremo del <association symbol> y la <association end area> más próxima al extremo correspondiente del <association symbol> es importante para esa <linked type reference area>."

*Sustitúyase* la regla <association end area> por:

```
<association end area> ::=
    { [ <role name> ] [ <multiplicity> ] [ <ordering area> ] [ <symbolic visibility> ] } set
```

y en los párrafos después de la sintaxis, *introdúzcanse* los siguientes cambios:

*Sustitúyase* "Si una <association end area> identifica" por "Si una <association end area> corresponde a".

*Sustitúyase* "está conectada a una <linked type reference area>" por "está vinculada a una <linked type reference area>".

## 1.10 Corrección de deficiencias – 9 – Gramática concreta – Sintaxis incorrecta

En la regla <agent body area> la opción <on exception association area> debe ser independiente de cualquier <start area>.



Suprímase "[ " antes de "[ <on exception association area>" e insértese "[ " antes de <start area> para armonizar la línea de sintaxis:

{ [<on exception association area>] [ <start area> ]

### 1.11 Corrección de deficiencia – 9 – Gramática concreta – Sustitución de "connected" por "attached"

En la regla <create line area>, *sustitúyase "connected" por "attached"* porque el (tipo) agente o partición de estado lo genera la regla del diagrama que los encierra.

### 1.12 Aclaración/corrección de deficiencia – 9 – Semántica – Máquina de estado predeterminada

Se pretende definir más claramente la máquina de estados para el caso en el que no hay otra máquina de estados explícita o implícita. La palabra "no" no se omitió por error antes de "ejemplares iniciales contenidos" en la segunda frase:

Sustitúyase el párrafo:

"Si un agente no tiene una máquina de estados explícita o implícita, pasa a una condición de parada tan pronto como se crean todos los agentes contenidos iniciales. Por lo tanto, un agente con ejemplares iniciales contenidos y sin máquinas de estado contenidas cesa de existir tan pronto como es creado."

*Por*

"Si un agente no tiene una máquina de estados explícita o implícita, existe una máquina de estados que sólo tiene un *Stop-node*. Tan pronto como se crean todos los agentes contenidos iniciales, el agente pasa a una condición de parada. Por lo tanto, un agente que no contiene ejemplares iniciales ni tampoco máquinas de estado contenidas cesa de existir tan pronto como es creado."

### 1.13 Corrección de deficiencias – 9.2 – Gramática concreta – Sintaxis incorrecta

En la regla <block diagram>, los <external channel identifiers> están *associated with* (asociados con) el bloque (y no "*connected to*" (conectados) al mismo).

Sustitúyase la línea de sintaxis:

*is connected to* { {<gate on diagram> | <external channel identifiers>}\* } *set*

*por*

*is connected to* { {<gate on diagram>}\* } *set*  
*is associated with* { <external channel identifiers>\* } *set*

### 1.14 Corrección de deficiencias – 9.3 – Gramática concreta – Sintaxis incorrecta

En la regla <process diagram>, los <external channel identifiers> están *associated with* (asociados con) el proceso (y no "*connected to*" (conectados) al mismo).

Sustitúyase la línea de sintaxis:

*is connected to* { {<gate on diagram> | <external channel identifiers>}\* } *set*

*por*

*is connected to* { {<gate on diagram>}\* } *set*  
*is associated with* { <external channel identifiers>\* } *set*

### 1.15 Aclaración – 9.5 – Gramática concreta – Sintaxis incorrecta

En la regla <procedure body area>, el {} **set** se *aplica* a todo el lado derecho para indicar más claramente que el orden no es importante. Así pues, la línea de sintaxis debe quedar de la siguiente forma:

```
{[ <on exception association area> ] [<procedure start area>]
  {<state area> | <exception handler area> | <in connector area> } * } set
```

### 1.16 Corrección de deficiencias – 10.1, 10.2 – Gramática concreta – Sustitución de "connected" por "attached"

La regla <channel definition area> está **connected to** (conectada a) los agentes, estados o puertas en cualquier extremo de canal. Ahora bien, **connected to** significa que el elemento de sintaxis se genera y, por consiguiente, es incorrecto. El metasímbolo correcto es **attached to** (vinculado). Sustitúyase la palabra "**connected**" por "**attached**" en la sintaxis. Así pues, la sintaxis modificada debe quedar de la siguiente manera:

```
<channel definition area> ::=
  <channel symbol>
  is associated with
  { [<channel name>] { [<signal list area>] [<signal list area>] } set }
  is attached to {
    { <agent area> | <state partition area> | <gate on diagram> }
    { <agent area> | <state partition area> | <gate on diagram> } } set
```

Por consiguiente, sustitúyase "conectado" por "vinculado" en el resto del texto de 10.1 (14 veces) y en 10.2 (7 veces después de haber suprimido lo indicado más adelante).

Así mismo en 10.1 *Gramática concreta*, sustitúyase la palabra "conexión" por "vinculación" (dos veces).

Al revisar estos cambios se observó que el texto de *Semántica* y *Modelo* en 10.2 menciona canales internos a la unidad de ámbito con la que están vinculados los <external channel identifiers>.

Por consiguiente *suprímase* el texto en la *semántica* 10.2:

"Cada canal identificado por un <channel identifier> en un <external channel identifiers> debe definirse en el mismo agente en el que se define la conexión y debe tener en la frontera de dicho agente uno de sus puntos extremos."

El siguiente texto de la *semántica* en 10.2 parece ser una restricción innecesaria y se ha sugerido su supresión:

"Cada canal definido en el agente circundante y cuyo entorno sea uno de sus puntos extremos, debe ser mencionado en un solo <external channel identifiers>."

El título *Semántica* en 10.2 es incorrecto (el texto se refiere básicamente a la gramática) y por lo tanto ha de *suprimirse*.

El texto en el apartado *Modelo* en 10.2 "en sus respectivas unidades de ámbito" ha de *suprimirse* porque es irrelevante: sólo hay una unidad de ámbito pertinente (la circundante).

Al final del texto de *Modelo* en 10.2 *añádase* la siguiente aclaración:

"Las identidades de los canales y de las puertas se obtienen a partir de la vinculación (véase 10.1)."

### 1.17 Aclaración – 11.2 – Gramática concreta – Sintaxis incorrecta

*Suprímase* la regla <connect association area>. Está definida en 11.11.4 que sustituye <connect area>. Véase además el cambio que se introduce a 11.11.4 en 1.33 de este corrigendum.

### 1.18 Aclaración – 11.2 – Gramática concreta – Utilización de <composite state name>

Se mejora la utilización de <composite state name>.

*Sustitúyase* "En este caso la <state area> sólo debe contener un <composite state name>" *por* "En este caso la <state area> sólo debe contener un <composite state name>".

### 1.19 Aclaración – 11.2 – Modelo – Utilización de <composite state name>

Se mejora la utilización de <composite state name>.

*Sustitúyase* "uno para cada <state name> y <composite state name> del cuerpo en cuestión" *por* "uno para cada <state name> del cuerpo en cuestión".

### 1.20 Corrección de deficiencias – 11.3 – Gramática concreta – Sintaxis incorrecta

La sintaxis de <input area>

```
<input symbol> contains { [<virtuality>] <input list> }  
[ is connected to <on exception association area> ]  
[ is associated with <solid association symbol> is connected to <enabling condition area> ]  
is followed by <transition area>
```

significa que cuando se emplea una condición habilitadora, debería haber una línea (un símbolo de asociación sólido) unida a la condición habilitadora Y una línea de flujo unida a la transición desde el símbolo de entrada. Evidentemente, esto es incorrecto. La sintaxis *correcta* es:

```
<input symbol> contains { [<virtuality>] <input list> }  
[ is connected to <on exception association area> ]  
{  
    | is connected to <enabling condition association area>  
    | is followed by <transition area> }  
}
```

además, *añádase* el siguiente párrafo después de esta sintaxis:

"La <enabling condition association area> define la <transition area> en el caso de una condición habilitadora."

Se ha de introducir el cambio correspondiente en 11.6 (véase 1.21 a continuación).

### 1.21 Corrección de deficiencias – 11.6 – Gramática concreta – Sintaxis incorrecta

Para corregir la sintaxis de <input area> (véase 1.20 anterior) se introduce una regla de sintaxis para <enabling condition association area> y se amplía la regla <enabling condition area> para incluir una <transition area>. La sintaxis *correcta* es:

```
<enabling condition association area> ::=  
    <solid association symbol> is connected to <enabling condition area>  
<enabling condition area> ::=  
    <enabling condition symbol> contains <provided expression>  
    is followed by <transition area>
```

además, *añádase* el siguiente párrafo después de esta sintaxis:

"La <transition area> corresponde a *Transition* del *Input-node* o *Spontaneous-transition* para la *Provided-expression*. Se incluye aquí la sintaxis para obtener la producción gráfica correcta de una línea de flujo desde el <enabling condition symbol> hacia la transición."

### 1.22 Corrección de deficiencias – 11.9 – Gramática concreta – Sintaxis incorrecta

La sintaxis de <spontaneous transition area> significa que cuando se emplea una condición habilitadora debería haber una línea (un símbolo de asociación sólido) que una la condición habilitadora Y una línea de flujo unida a la transición desde el símbolo de entrada. Evidentemente, esto es incorrecto. La sintaxis *correcta* es:

```
<input symbol> contains { [<virtuality>] <spontaneous designator> }  
[ is connected to <on exception association area> ]  
{  
    | is connected to <enabling condition association area>  
    | is followed by <transition area> }
```

además, *añádase* el siguiente párrafo después de la sintaxis:

"La <enabling condition association area> define la <transition area> en caso de una condición habilitadora."

Hay que introducir el correspondiente cambio en 11.6 (véase 1.21 anterior).

### 1.23 Aclaración – 11.11 – Utilización de <composite state name>

Se mejora la utilización de <composite state name>.

En el segundo párrafo, *sustitúyase* "<composite state name>" *por* "<composite state name>".

### 1.24 Aclaración – 11.11 – Gramática abstracta – Procedimiento con estados para procedimientos de entrada/salida

El texto después de la sintaxis abstracta que comienza por "*Entry-procedure-definition* representa" antes de "*Semántica*" se *sustituye por*:

"Un procedimiento con estados es un procedimiento que contiene un estado (explícito o implícito) o llamadas a un procedimiento con estados.

*Entry-procedure-definition* de un *Composite-state-graph* o *State-aggregation-node* es un procedimiento sin parámetros definidos explícitamente en el *Composite-state-graph* o *State-aggregation-node* correspondiente a la entrada del nombre. El procedimiento de entrada no debe ser un procedimiento con estados.

*Exit-procedure-definition* de un *Composite-state-graph* o *State-aggregation-node* es un procedimiento sin parámetros definidos explícitamente en el *Composite-state-graph* o *State-aggregation-node* correspondiente a la salida del nombre. El procedimiento de salida no debe ser un procedimiento con estados.

### 1.25 Corrección de deficiencias – 11.11.1 – Gramática concreta – Sintaxis incorrecta

En la regla <composite state graph area>, *sustitúyase* "*associated with*" antes de { <state connection point>\* } *set* por "*connected to*" porque los elementos están unidos (como puertas) y el texto los describe además como si estuvieran "conectados". Asimismo, cada vez que aparezca <state connection point> *sustitúyase por* <state connection point area> para ser coherente con la gramática gráfica.

### 1.26 Aclaración – 11.11.1 – Gramática concreta – Utilización de <composite state name>

Se ha mejorado la utilización de <composite state name>.

En la regla <composite state heading>, *sustitúyase* "<composite state name>" *por* "<state name>".

### 1.27 Aclaración – 11.11.1 – Gramática concreta – Sintaxis incorrecta

En la regla <composite state body area> se debe *aplicar* {} *set* a todo lado derecho para aclarar que el orden no es importante. Además, la distribución se hace coherente con reglas similares. La sintaxis resultante es:

```
{ [<on exception association area>] <start area>*  
  { <state area> | <exception handler area> | <in connector area> }* } set
```

### 1.28 Corrección de deficiencias – 11.11.2 – Gramática concreta – Sintaxis incorrecta

En la regla <state aggregation area>, *sustitúyase* "*associated with*" antes de { <state connection point>\* } *set por* "*connected to*" porque los elementos están unidos (como puertas) y el texto los describe además como si estuvieran "conectados". Por otra parte cada vez que aparezca <state connection point>, *sustitúyase por* <state connection point area> para ser coherente con la gramática gráfica.

### 1.29 Aclaración – 11.11.2 – Gramática concreta – Utilización de <composite state name>

Se ha mejorado la utilización de <composite state name>.

En la regla <state aggregation heading>, *sustitúyase* "<composite state name>" *por* "<state name>".

### 1.30 Corrección de deficiencias – 11.11.2 – Gramática concreta – Permite puertas en la partición de estado heredada

Debería permitirse que hubiera puertas dentro de la <inherited state partition definition>. La regla revisada es:

```
<inherited state partition definition> ::=  
  <dashed state symbol> contains { <composite state identifier> { <gate>* } set }
```

### 1.31 Corrección de deficiencia – 11.11.2 – Gramática concreta – Sintaxis incorrecta

En la regla <state partition connection area> se empleó por error [] en lugar de {}. No debería permitirse que los símbolos de asociación sólidos estén desconectados.

Además, <outer graphical point> se mostró por error asociado a <frame symbol>, mientras que <frame symbol> es una producción sintáctica del (tipo de) agregación de estado envolvente (una <state aggregation area> o <composite state type diagram>). El <outer graphical point> tiene que ser definido por la <state connection point area> del diagrama.

```
<state partition connection area> ::=  
  <solid association symbol>  
  is attached to <frame symbol>  
  is attached to <state partition area>  
  is connected to { <outer graphical point> <inner graphical point> }
```

además, *añádase* el siguiente párrafo:

"El <solid association symbol> está vinculado a un extremo del <frame symbol> del diagrama envolvente y el <outer graphical point> se ubica cerca del exterior de este <frame symbol> del diagrama envolvente. El <solid association symbol> está vinculado en el otro extremo a una <state partition area> y el <inner graphical point> se ubica cerca del mismo. El <outer graphical point> hará referencia únicamente a los nombres definidos como puntos de entrada o salida del estado del diagrama envolvente. El <inner graphical point> hará referencia únicamente a los nombres definidos como puntos de entrada o salida del estado de la <state partition area>."

*Suprímase* las partes gráficas correspondientes a las reglas del punto gráfico (que son sintaxis inválidas de acuerdo con la metagramática) de modo que resulten ser lo siguiente:

```
<outer graphical point> ::=  
    { <state entry points> | <state exit points> }  
<inner graphical point> ::=  
    { <state entry points> | <state exit points> }
```

### 1.32 Corrección de deficiencias – 11.11.3 – Gramática concreta – Sintaxis incorrecta

*Suprímase* la última línea "*is connected to* <frame symbol>" de la regla <state connection point> porque el punto de conexión está conectado a la trama y sintácticamente produce un punto de conexión (en lugar de lo contrario). Además, cada vez que aparezca <state connection point> *sustitúyase por* <state connection point area> *para* ser coherente con la gramática gráfica.

### 1.33 Aclaración – 11.11.4 – Gramática concreta – Sintaxis incorrecta

*Sustitúyase* la regla <connect area> *por*:

```
<connect association area> ::=  
    <solid association symbol> is associated with { [<virtuality>] [<connect list>] }  
    [ is connected to <on exception association area> ]  
    is followed by <exit transition area>
```

y cada vez que aparezca <connect area> *sustitúyase por* <connect association area>. Véanse además los cambios introducidos en 11.2 en 1.17 de este corrigendum.

### 1.34 Aclaración – 11.12.2.2 – Gramática concreta – Sintaxis incorrecta

*Sustitúyase* *connected* por *attached* en la regla <merge area> para garantizar que existe otra producción que produce la línea de flujo.

### 1.35 Aclaración – 11.13.3 – Gramática concreta – Ambigüedad

En <procedure call body>, las alternativas <procedure identifier> y <procedure type expression>, son ambiguas porque un <procedure identifier> es una <procedure type expression>. La sintaxis correcta es la siguiente:

```
<procedure call body> ::=  
    [ this ] <procedure type expression> [<actual parameters>]
```

Para explicar el significado de <procedure identifier> en el resto del texto de esta cláusula, *añádase* el siguiente párrafo después de esta sintaxis:

"En el texto que figura a continuación, <procedure identifier> significa el <procedure identifier> para el <base type> de la <procedure type expression>, y si la <procedure type expression> es una <procedure identifier>, se trata simplemente de este <procedure identifier>."

### 1.36 Modificación – 11.13.4 – Semántica – Encaminamiento de señalización

Para poder dirigir una señal hacia un ejemplar de agente haciendo referencia a su Pid y permitir que el agente encamine internamente la señal hacia el agente contenido correcto, la interfaz de un agente debería incluir la comunicación con agentes contenidos. Para ello también es necesario hacer el cambio indicado en 1.37 siguiente.

Insértese el siguiente párrafo antes del párrafo que comienza por "Obsérvese que el hecho de que se especifique el mismo *Channel-identifier*":

"Cuando un ejemplar de señal se entrega a un ejemplar de un conjunto de ejemplares de agentes y hay un trayecto de comunicación interno que transporta la señal a la máquina de estados del ejemplar del agente, el ejemplar de la señal se entrega a dicha máquina. De lo contrario se elige al

azar un trayecto de comunicación dentro del ejemplar del agente con la capacidad para transportar el ejemplar de la señal, y éste se entrega al conjunto de ejemplares de un agente contenido."

### **1.37 Modificación – 12.1.2 – Modelo de interfaces implícitas**

Para poder dirigir una señal hacia un ejemplar de agente haciendo referencia a su Pid y permitir que el agente encamine internamente la señal hacia el agente contenido correcto, la interfaz de un agente debería incluir la comunicación con agentes contenidos. El texto que define la interfaz implícita no se entiende muy bien y, por esa razón, se desea perfeccionarlo.

*Sustitúyase el párrafo:*

"La interfaz definida por un agente o tipo de agente contiene en su <interface specialization> todas las interfaces dadas en la lista de señales entrantes asociadas con puertas explícitas o implícitas del agente o tipo de agente, de manera que las puertas están conectadas mediante canales implícitos o explícitos a puertas de la máquina de estados del agente o tipo de agente. La interfaz también contiene en su <interface use list> todas las señales, variables remotas y procedimientos remotos dados en la lista de señales entrantes asociadas con puertas explícitas o implícitas del agente o tipo de agente de manera que las puertas están conectadas mediante canales implícitos o explícitos a puertas de la máquina de estados del agente o tipo de agente. Además, la interfaz para un tipo de agente que hereda otro tipo de agente contiene también en su <interface specialization> la interfaz implícita definida por el tipo de agente heredado."

*Por el siguiente párrafo:*

"Las puertas de un agente (o tipo de agente) conectadas internamente son puertas explícitas o implícitas del agente (o del tipo de agente, respectivamente) que está conectado mediante canales implícitos o explícitos a las puertas de la máquina de estados del agente (o tipo de agente, respectivamente) o a un agente contenido. La interfaz definida por el agente o el tipo de agente contiene en su <interface specialization> todas las interfaces que figuran en la lista de señales entrantes correspondiente a las puertas conectadas internamente. La interfaz contiene en su <interface use list> todas las señales, variables remotas y procedimientos remotos que figuran en la lista de señales entrantes correspondiente a las puertas conectadas internamente. Además la interfaz de un tipo de agente que hereda otro tipo de agente también contiene en su <interface specialization> la interfaz implícita definida por el tipo de agente heredado."

Si el texto de este párrafo "la máquina de estados del agente (o tipo de agente, respectivamente) o a un agente contenido" se sustituye por "la máquina de estados del agente (o tipo de agente, respectivamente)" debería tener el mismo significado que antes.

Por otra parte, es necesario modificar el párrafo a continuación que define la interfaz implícita de una máquina de estados del agente para excluir los elementos que forman parte únicamente de los agentes contenidos. Por esa razón se ha tratado de mejorar este texto, en particular la descripción de los elementos que no son pertinentes a la comunicación a través de las puertas del agente.

*Sustitúyase el texto original:*

"La interfaz definida por la máquina de estados de un agente o tipo de agente contiene en su <interface specialization> la interfaz definida por el mismo agente o tipo de agente. Además, la interfaz contiene en su <interface specialization> todas las interfaces dadas en la lista de señales entrantes asociadas con puertas explícitas o implícitas de la máquina de estados, de manera que las puertas no están conectadas mediante canales implícitos o explícitos a puertas explícitas o implícitas del agente o tipo de agente. La interfaz contiene también en su <interface use list> todas las señales, variables remotas y procedimientos remotos dados en la lista de señales entrantes asociadas con puertas explícitas o implícitas de la máquina de estados, de modo que las puertas no están conectadas por canales implícitos o explícitos a puertas explícitas o implícitas del agente o tipo de agente. Si la entidad contenedora es un tipo de agente que hereda otro tipo de agente, la

interfaz contendrá también en su <interface specialization> la interfaz implícita de la máquina de estados del tipo de agente heredado."

Por el siguiente texto:

"La interfaz definida por la máquina de estados de un agente o tipo de agente contiene en su <interface specialization> la interfaz definida por el mismo agente o tipo de agente, salvo las partes de esa interfaz que sólo conciernen a los agentes contenidos. Sin embargo, la interfaz también contiene en su <interface specialization> todas las interfaces que figuran en la lista de señales entrantes asociada a las puertas explícitas o implícitas de la máquina de estados. Asimismo la interfaz contiene en su <interface use list> todas las señales, variables, remotas y procedimientos remotos que figuran en la lista de señales entrantes asociada con las puertas explícitas o implícitas de la máquina de estados. Si la entidad contenedora es un tipo que hereda otro tipo de agente, la interfaz podrá contener también en su <interface specialization> la interfaz implícita de la máquina de estados del tipo de agente heredado."

Este texto puede modificarse para obtener el mismo significado del texto original con sólo suprimir "salvo las partes de esa interfaz que sólo conciernen a los agentes contenidos".

### 1.38 Corrección de deficiencias – 12.1.8 – Gramática concreta – Sintaxis incorrecta

En la regla <operation body area>, el <procedure start area> debería ser opcional. Se añaden "[ " y " ]" antes y después de <procedure start area>. De este modo en la definición perfeccionada o abstracta se puede omitir la <procedure start area>. {} *set* se *aplica* a toda la parte derecha para aclarar que el orden no es importante. La línea de sintaxis resultante es:

```
{ [ <on exception association area> ] [ <procedure start area> ]  
  { <in connector area> | <exception handler area> } * } set
```

### 1.39 Aclaración – 12.3.4.1 – Semántica – Unidad de tiempo

Añádase la siguiente frase:

"A no ser que se especifique lo contrario, la unidad de tiempo en las especificaciones SDL es de 1 segundo", en los siguientes lugares:

- En el apartado *Semántica* de 12.3.4.1 después del texto "la unidad de tiempo depende del sistema".
- En el apartado Utilización de D.3.11.2 para el género duración al final del párrafo justo antes del D.3.12.
- En el apartado de Utilización de D.3.12.2 para el género tiempo al final del párrafo justo antes del D.3.13.

### 1.40 Aclaración – 13.1 – Gramática concreta – Sintaxis perfeccionada

Se vuelve a *definir* la regla <option symbol>:

```
<option symbol> ::=  
  {  
    <dashed line symbol> is attached to <dashed line symbol>  
    <dashed line symbol> is attached to <dashed line symbol>  
    <dashed line symbol> is attached to <dashed line symbol>  
    { <dashed line symbol> is attached to <dashed line symbol> }+ } set
```

además, *insértese* la palabra "rectilíneo" antes de la palabra "polígono" en la frase siguiente a esta regla.

### 1.41 Aclaración – D.3.11.2, D.3.12.2 – Unidad de tiempo

Véase 1.39 de este corrigendum.



## 2 Modificación del B.9 (anexo B, "Compatibilidad con sistemas anteriores")

La edición anterior no incluía algunos de los elementos de gramática necesarios para los operadores definidos mediante la sintaxis SDL-92. La nueva cláusula B.9 que se presenta a continuación sustituye la que figura en la enmienda 1 a la Z.100 (10/2003) e incluye el texto anterior.

### B.9 Comportamiento de las operaciones

#### *Gramática concreta*

Para garantizar la compatibilidad con los modelos SDL-92, se añade sintaxis adicional para las definiciones de operador: <legacy operator definition>, <legacy operator reference> y <legacy external operator definition>. Se utilizan éstas en lugar de <operation definitions> dentro de una <legacy data type definition> (véase B.12).

```
<legacy operator definition> ::=
    {<package use clause>}*
    <operation heading> <end>
    { <entity in operation> }*
    <start>
    endoperator
    [ {<operation identifier> | <operation name> } ] <end>
<legacy operator reference> ::=
    <operation heading> referenced <end>
<legacy external operator definition> ::=
    operator <operation name> [ <legacy procedure signature> ] external <end>
```

Una <operation heading> en una <legacy operator definition> o <legacy operator reference> utilizará el **operador** clave.

Una <legacy operator definition> corresponde a una <operation definition> en SDL-2000.

Una <legacy operator reference> corresponde a una <operation reference> en SDL-2000.

Una <legacy external operator definition> corresponde a una <external operation definition> en SDL-2000.

<start> se define en la Rec. UIT-T Z.106. El cuerpo de la transición para <start> deberá contener sólo los elementos que se permiten en una definición de operación.

La sintaxis de <formal operation parameters> se ha ampliado para permitir que los parámetros formales puedan especificarse con **fpar**.

```
<formal operation parameters> ::=
    ( <operation parameters> {, <operation parameters> }* )
    | [ <end> ] fpar <operation parameters> {, <operation parameters> }*
```

NOTA – La opción <end> antes de la palabra clave **fpar** se ha añadido para validar modelos que se han de definir utilizando herramientas que requieren un punto y coma en este punto aunque no fuese válido en SDL/GR en SDL-92.

La sintaxis de <operation result> se ha ampliado para permitir la especificación de **returns**.

```
<operation result> ::=
    <result sign> [ <variable name> ] <sort>
    | returns [ <variable name> ] <sort>
```

## 3 Adiciones al anexo B – Compatibilidad con sistemas anteriores

La gramática definida en el anexo B que figura en la enmienda 1 a la Z.100 (10/2003) no incluye algunos elementos de gramática necesarios para utilizar los datos y tipos de datos que emplean la sintaxis SDL-92. Se ha *añadido* lo siguiente al anexo B para soportar la sintaxis SDL-92.

## B.11 Definición de datos

### *Gramática concreta*

Para lograr la compatibilidad con los modelos SDL-92, se ha ampliado la sintaxis para que <legacy data type definition> (véase B.12) y <legacy syntype definition> (véase B.13) puedan aparecer en una <data definition>.

```
<data definition> ::=  
    <data type definition>  
    | <legacy data type definition>  
    | <interface definition>  
    | <syntype definition>  
    | <legacy syntype definition>  
    | <synonym definition>
```

Para ser coherente con el SDL-92, el <sort> en cualquiera de los componentes de una <data definition> siempre debe ser un identificador de género o syntype.

## B.12 Definición de tipo de datos

### *Gramática concreta*

Para lograr la compatibilidad con modelos SDL-92, se ha ampliado la sintaxis para que pueda utilizarse la <legacy syntype definition> (véase B.13) en el <entity in data type>.

```
<entity in data type> ::=  
    <data type definition>  
    | <legacy data type definition>  
    | <syntype definition>  
    | <legacy syntype definition>  
    | <synonym definition>  
    | <exception definition>  
<legacy data type definition> ::=  
    newtype <sort name>  
        [ <formal context parameters> ]  
        [ <data type specialization> ]  
        | <legacy generators>  
        | <structure definition> ]  
        [ <literal list> ]  
        [ <legacy operator signatures> ]  
        {  
            <legacy operator definition>  
            | <legacy operator reference>  
            | <legacy external operator definition> } *  
        [ <default initialization> [ <end> ] ]  
        [ constants <range condition> ]  
    endnewtype [ <sort name> ]
```

Para ser coherente con el SDL-92 la <data type specialization> en una <legacy data type definition> debe contener una <legacy data inheritance> (véase B.8).

Para ser coherente con el SDL-92 la <structure definition> en una <legacy data type definition> no debe contener <visibility>, **optional** o <field default initialisation>.

Para ser coherente con el SDL-92 la <literal list> en una <legacy data type definition> no debe contener <visibility> o <named number>.

Si una definición <legacy data type definition> contiene una <range condition>, ésta representa la definición de syntype y un tipo de datos progenitor anónimo.

La definición de un operador tradicional debe definirse en la <legacy operator definition> o en el operador al que se hace referencia mediante una <legacy operator reference> o <legacy external operator definition> (véase B.9).

### B.12.1 Generadores

#### *Gramática concreta*

Aunque el SDL-2000 no incluye generadores para tipos de datos, los tipos de datos parametrizados en **package** predefinidos en SDL-2000 sustituyen los generadores tales como Array que fueron incluidos en el lote predefinido en SDL-92. La <legacy data type definition> incluye la sintaxis de <legacy generators> de modo que puedan utilizarse estos tipos de datos parametrizados.

```
<legacy generators> ::=
    <sort identifier> ( <legacy generator actual> { , <legacy generator actual> }* )
<legacy generator actual> ::=
    <sort>
    | <literal signature>
    | <operator name>
    | <constant expression>
```

El <sort identifier> debería de identificar uno de los tipos de datos parametrizados en el **package** predefinido. El <legacy generator actual> debería ser un parámetro actual adecuado para los tipos de datos parametrizados.

### B.12.2 Firmas de operador

#### *Gramática concreta*

Para lograr la compatibilidad con los modelos SDL-92 una alternativa son las <legacy operator signatures>.

```
<legacy operator signatures> ::=
    operators
    <legacy operator signature> { <end> <legacy operator signature> }* [ <end> ]
<legacy operator signature> ::=
    <operator name> : <arguments> -> <sort>
```

El <legacy operator signature> representa una *Operation-Signature*.

El <sort> de una <legacy operator signature> representa el *Result* de la *Operation-Signature*.

### B.13 Syntypes

#### *Gramática concreta*

Para lograr la compatibilidad con modelos SDL-92, una alternativa es la <legacy syntype definition>.

```
<legacy syntype definition> ::=
    syntype
    <syntype name> = <parent sort identifier>
    [ <default initialization> [ <end> ] ]
    [ constants <range condition> ]
    endsyntype [ <syntype name> ]
```

Véase también la <legacy data type definition> para **syntype** combinado con **newtype** en B.12.





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

Serie A	Organización del trabajo del UIT-T
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	Redes de cable y transmisión de programas radiofónicos y televisivos, 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	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
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, comunicaciones de sistemas abiertos y seguridad
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
<b>Serie Z</b>	<b>Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación</b>