



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**Z.100**

**Corrigendum 1**  
(10/2001)

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 – 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 combinada arborescente y tabular	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**

Este corrigendum contiene una lista de cambios de la Rec. UIT-T Z.100 (Lenguaje de especificación y descripción) que fue aprobada por la Comisión de Estudio 10 en noviembre de 2000, que incluye cambios adicionales acordados por el Grupo de Expertos en reuniones subsiguientes, y la lista final acordada por la CE 10 en la reunión de septiembre de 2001.

#### **Orígenes**

El corrigendum 1 a la Recomendación UIT-T Z.100, preparado por la Comisión de Estudio 10 (2001-2004) del UIT-T, fue aprobado por el procedimiento de la Resolución 1 de la AMNT el 29 de octubre 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

		<b>Página</b>
1	Objetivos y alcance.....	1
2	Terminología .....	1
3	Mantenimiento de las Recomendaciones UIT-T Z.100 a Z.109 .....	1
4	Cambios de la Recomendación Z.100 .....	2
4.1	Error material – Subrayado de <Natural literal name> .....	2
4.2	Ampliación – Añádase el <cuerpo de comentario> antes de <left curly bracket>.....	2
4.3	Error material – <end> – Anexo D.....	2
4.4	Error material – Palabra clave de enunciado de bucle – Cláusula 1.5: Diferencias...	2
4.5	Aclaración – Cláusula 5.3.2: Ítems de enumeración titulados, Modelo.....	2
4.6	Error material y clarificación – Cláusula 5.4.2: BNF.....	2
4.7	Aclaración – Ejemplo – Cláusula 5.4.3: Metalenguaje para gramática gráfica .....	2
4.8	Error material – Símbolos formados por líneas – Cláusula 5.4.3: Metalenguaje para gramática gráfica .....	3
4.9	Modificación – Sintaxis de comentario – Cláusula 6.1: Reglas léxicas.....	3
4.10	Ampliación – Señal de intervalo – Cláusula 6.1: Reglas léxicas .....	4
4.11	Error material – Palabra clave del enunciado de bucle – Cláusula 6.1: Reglas léxicas .....	4
4.12	Error material – Referencia – Cláusula 6.3: Reglas de visibilidad, nombres e identificadores, Gramática textual concreta .....	4
4.13	Errores materiales – Contenido de los dibujos – Cláusula 6.6: Partición de dibujos.	4
4.14	Modificación – Cambio de la sintaxis para comentario – Cláusula 6.7: Comentario	5
4.15	Error material – Gramática para <specification area> – Cláusula 7.1: Marco.....	5
4.16	Error material – Permite señales al entorno para sistemas basados en el tipo – Cláusula 7.1: Marco.....	5
4.17	Aclaración – Tipo ejemplificado potencialmente – Cláusula 7.1: Marco, Semántica .....	5
4.18	Aclaración – Sintaxis gráfica concreta para <diagram in package> – Cláusula 7.2: Lote.....	5
4.19	Error material – Gramática para <specification area> – Cláusula 7.2: Lote, Gramática textual concreta .....	6
4.20	Error material – Unión de lotes – Cláusula 7.2: Lote, Modelo .....	6
4.21	Aclaración – Cláusula 8.1.1.1: Tipo de agente, Gramática abstracta.....	6
4.22	Aclaración – Sintaxis simplificada – Cláusula 8.1.1.1: Tipo de agente, Gramática textual concreta.....	6
4.23	Aclaración – Sintaxis simplificada – Cláusula 8.1.1.1: Tipo de agente, Gramática gráfica concreta .....	6

4.24	Error material – Definición de conjunto de salida completo – Cláusula 8.1.1.1: Tipo de agente, Semántica.....	7
4.25	Aclaración – Modelo de tipo de agente – Cláusula 8.1.1.1: Tipo de agente, Modelo.....	7
4.26	Error material – Faltan parámetros de contexto – Cláusula 8.1.1.2: Tipo de sistema, Gramática textual concreta.....	7
4.27	Error material – Sintaxis simplificada – Cláusula 8.1.1.2: Tipo de sistema, Gramática textual concreta.....	7
4.28	Error material – Puertas/sintaxis omitidas, simplificadas – Cláusula 8.1.1.2: Tipo de sistema, Gramática gráfica concreta.....	8
4.29	Aclaración – Cláusula 8.1.1.2: Tipo de sistema, Gramática gráfica concreta.....	8
4.30	Aclaración – Simplificación de sintaxis – Cláusula 8.1.1.3: Tipo de bloque, Gramática textual concreta.....	8
4.31	Error material – Faltan parámetros de contexto – Cláusula 8.1.1.3: Tipo de bloque, Gramática textual concreta.....	8
4.32	Aclaración – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta.....	8
4.33	Aclaración – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta.....	8
4.34	Corrección de error – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta.....	8
4.35	Aclaración – Simplificación de la sintaxis – Cláusula 8.1.1.4: Tipo de proceso, Gramática textual concreta.....	9
4.36	Error material – Faltan parámetros de contexto – Cláusula 8.1.1.4: Tipo de proceso, Gramática textual concreta.....	9
4.37	Aclaración – Cláusula 8.1.1.4: Tipo de proceso, Gramática gráfica concreta.....	9
4.38	Aclaración – Cláusula 8.1.1.4: Tipo de proceso, Gramática gráfica concreta.....	9
4.39	Corrección de error – Cláusula 8.1.1.4: Tipo de proceso, Semántica.....	9
4.40	Error material – Alineación con el anexo F – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática abstracta.....	9
4.41	Corrección de error – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática abstracta.....	10
4.42	Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta.....	10
4.43	Error material – Faltan parámetros de contexto – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta.....	10
4.44	Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta.....	10
4.45	Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática gráfica concreta.....	10
4.46	Corrección de error – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática gráfica concreta.....	10

4.47	Error material – Tipo subrayado – Cláusula 8.1.3.1: Definición de sistema basada en el tipo de sistema, Gramática textual concreta .....	11
4.48	Error material – Transformación innecesaria – Cláusula 8.1.3.1: Definición de sistema basada en el tipo de sistema.....	11
4.49	Error material – Transformación innecesaria – Cláusula 8.1.3.2: Definición de bloque basada en tipo de bloque.....	11
4.50	Error material – Transformación innecesaria – Cláusula 8.1.3.3: Definición de proceso basada en tipo de proceso.....	12
4.51	Error material – Falta exported – Cláusula 8.1.4: Tipo abstracto, Gramática concreta.....	12
4.52	Error material – Sort en propiedad de parámetro de señal – Cláusula 8.1.5: Referencias de tipo, Gramática textual concreta .....	12
4.53	Error material – Falta exported – Cláusula 8.1.5: Referencias de tipo, Gramática gráfica concreta .....	13
4.54	Error material – Constricción de punto extremo demasiado restrictiva – Cláusula 8.1.6: Puerta, Gramática textual concreta.....	13
4.55	Error material – Identificador erróneo – Cláusula 8.2.1: Parámetros de contexto de tipo de agente, Gramática textual concreta.....	13
4.56	Aclaración – Significado del carácter dos puntos – Cláusula 8.2.2: Parámetros de contexto de agente, Gramática textual concreta .....	13
4.57	Aclaración – Enlace con sintaxis abstracta – Cláusula 8.3.1: Adición de propiedades, Gramática textual concreta.....	14
4.58	Error material – Referencia – Cláusula 8.3.3: Transición/conservación virtual, Gramática textual concreta .....	14
4.59	Error material – Referencia a anexo F – Cláusula 8.3.3: Transición/conservación virtual, Semántica .....	14
4.60	Error material – Faltan parámetros de contexto – Cláusula 8.4: Asociaciones, Gramática textual concreta .....	14
4.61	Aclaración – Cláusula 9: Agentes, Gramática abstracta.....	14
4.62	Aclaración – Cláusula 9: Agentes, Gramática textual concreta .....	14
4.63	Errores materiales – Cláusula 9: Agentes, Gramática textual concreta.....	15
4.64	Aclaración – Cláusula 9: Agentes, Gramática textual concreta .....	15
4.65	Aclaración – Simplificación – Cláusula 9: Agentes, Gramática textual concreta.....	15
4.66	Aclaración – Cómo se describe – Cláusula 9: Agentes, Gramática gráfica concreta.	15
4.67	Error material – Cómo se describe – Cláusula 9: Agentes, Gramática gráfica concreta.....	16
4.68	Error material – Restricción innecesaria en conexión por puerta – Cláusula 9: Agentes, Semántica .....	16
4.69	Aclaración – Condición de parada para los agentes iniciales – Cláusula 9: Agentes, Semántica .....	16

4.70	Error material – Conjunto completo de señales de entrada válidas – Cláusula 9: Agentes, Semántica .....	16
4.71	Error material – Plural incorrecto – Cláusula 9: Agentes, Semántica .....	17
4.72	Error material – Definición del conjunto de salida completo – Cláusula 9: Agentes, Semántica .....	17
4.73	Aclaración – Valor pid de agentes contenidos – Cláusula 9: Agentes, Modelo.....	17
4.74	Error material – Puertas omitidas – Cláusula 9.1: Sistema, Gramática gráfica concreta.....	17
4.75	Error material – Condición incorrecta – Cláusula 9.1: Sistema, Gramática gráfica concreta.....	17
4.76	Aclaración – Cláusula 9.2: Bloque, Gramática gráfica concreta.....	17
4.77	Aclaración – Cláusula 9.3: Proceso, Gramática gráfica concreta.....	17
4.78	Error material – Identificador de procedimiento de nodo/base de arranque – Cláusula 9.4: Procedimiento, Gramática abstracta.....	18
4.79	Error material – Falta exported – Cláusula 9.4: Procedimiento, Gramática textual concreta.....	18
4.80	Error material – Faltan parámetros de contexto – Cláusula 9.4: Procedimiento, Gramática textual concreta .....	18
4.81	Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta.....	18
4.82	Error material – Falta estado compuesto – Cláusula 9.4: Procedimiento, Gramática gráfica concreta .....	18
4.83	Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta.....	19
4.84	Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta.....	19
4.85	Error material – Cláusula 9.4: Procedimiento, Semántica .....	19
4.86	Error material – Referencia al anexo F – Cláusula 9.4: Procedimiento, Semántica...	19
4.87	Error material – Referencia al anexo F – Cláusula 10.1: Canal, Gramática textual concreta.....	19
4.88	Aclaración – Cómo se describió – Cláusula 10.1: Canal, Gramática gráfica concreta.....	20
4.89	Aclaración – Cómo se describió – Cláusula 10.1: Canal, Gramática gráfica concreta.....	20
4.90	Error material – Referencia al anexo F – Cláusula 10.1: Canal, Modelo.....	20
4.91	Error material – Cláusula 10.2: Conexión, Gramática gráfica concreta.....	22
4.92	Error material – Cláusula 10.2: Conexión, Modelo .....	22
4.93	Error material – Faltan parámetros de contexto – Cláusula 10.3: Señal, Gramática textual concreta.....	22
4.94	Aclaración – Cláusula 10.3: Señal, Gramática gráfica concreta .....	23
4.95	Error material – Falta interfaz – Cláusula 10.4: Definición de lista de señales, Gramática textual concreta .....	23

4.96	Error material – Falta interfaz – Cláusula 10.4: Definición de lista de señales, Gramática textual concreta .....	23
4.97	Error material – Falta excepción – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta .....	23
4.98	Error material – Definición del conjunto de salida completo – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta .....	23
4.99	Error material – Definición del conjunto de salida completo – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta .....	23
4.100	Errores materiales – Cambio de renglón suplementario, tabulador suplementario – Cláusula 10.5: Procedimientos remotos, Modelo .....	23
4.101	Error material – nodelay – Cláusula 10.5: Procedimientos remotos, Modelo.....	24
4.102	Error material – Falta excepción – Cláusula 10.6: Variables remotas, Gramática textual concreta.....	24
4.103	Error material – Referencia al anexo F y nodelay – Cláusula 10.6: Variables remotas .....	24
4.104	Error material – Error de presentación – Cláusula 11.2: Estado, Gramática abstracta.....	24
4.105	Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Sintaxis abstracta .....	25
4.106	Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Gramática textual concreta .....	25
	11.2.1 Estado básico.....	25
4.107	Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Gramática gráfica concreta.....	26
4.108	Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Semántica .....	26
4.109	Aclaración – Selección de transición – Cláusula 11.2.1 : Estado básico, Semántica.....	27
4.110	Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado básico, Modelo .....	27
4.111	Aclaración – <action statement> por <action> – Cláusula 11.3: Entrada, Modelo ...	27
4.112	Error material – Omisión de on-exception – Cláusula 11.5: Señal continua, Gramática abstracta .....	27
4.113	Aclaración – Selección de transición – Cláusula 11.5: Señal continua, Semántica ...	27
4.114	Aclaración – Selección de transición – Cláusula 11.6: Condición habilitadora, Semántica .....	28
4.115	Aclaración – Cláusula 11.7: Conservación (save), Gramática abstracta.....	28
4.116	Aclaración – body area en lugar de graph area – Cláusula 11.10: Etiqueta, Gramática textual concreta .....	28
4.117	Aclaración – <action statement> por <action> – Cláusula 11.10: Etiqueta, Gramática textual concreta .....	28

4.118	Aclaración – body area en lugar de graph area – Cláusula 11.10: Etiqueta, Gramática gráfica concreta.....	28
4.119	Aclaración – <action statement> por <action> – Cláusula 11.10: Etiqueta, Modelo	28
4.120	Aclaración – <terminator statement> por <terminator> – Cláusula 11.10: Etiqueta, Modelo.....	28
4.121	Error material – Falta "por" – Cláusula 11.11: Máquina de estados y estado compuesto.....	29
4.122	Aclaración – Cláusula 11.11: Máquina de estados y estado compuesto, Semántica .	29
4.123	Error material – Falta Modelo – Cláusula 11.11: Máquina de estados y estado compuesto, Modelo .....	29
4.124	Error material – Nodo de comienzo facultativo – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática abstracta.....	29
4.125	Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta.....	29
4.126	Error material – Especialización omitida – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta.....	30
4.127	Error material – Ítems omitidos – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta .....	30
4.128	Error material – Regla mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta.....	30
4.129	Aclaración – Oración mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta.....	30
4.130	Aclaración – Utilización de puerta – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta.....	30
4.131	Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta.....	31
4.132	Error material – Ítems omitidos – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta.....	31
4.133	Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta.....	31
4.134	Aclaración – Oración mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta.....	31
4.135	Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Semántica .....	32
4.136	Aclaración – Cláusula 11.11.2: Agregación de estado.....	32
4.137	Aclaración – Condiciones – Cláusula 11.11.2: Agregación de estado, Gramática abstracta.....	32
4.138	Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática abstracta.....	32
4.139	Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta..	32
4.140	Error material – Especialización omitida – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta.....	33
4.141	Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta..	33

4.142	Error material – Permitir empty body – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta .....	33
4.143	Aclaración – Referencia de estado compuesto – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta .....	33
4.144	Error material – Referencia de estado compuesto – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta .....	33
4.145	Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta..	33
4.146	Error material – Permitir empty body – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta .....	33
4.147	Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta..	34
4.148	Error material – Identificador – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta .....	34
4.149	Error material – Sintaxis de puntos de conexión – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta .....	34
4.150	Error material – <gate>s omitidas – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta .....	34
4.151	Aclaración – Condiciones – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta .....	35
4.152	Aclaraciones – Cláusula 11.11.2: Agregación de estado, Semántica .....	35
4.153	Aclaración – Oración mal colocada – Cláusula 11.11.3: Punto de conexión de estado, Semántica .....	35
4.154	Ampliación – Lista de conexión (connect list) más flexible – Cláusula 11.11.4: Connect .....	35
4.155	Error material – Tratamiento de excepción – Cláusula 11.12.1: Cuerpo de transición, Gramática textual concreta .....	36
4.156	Error material – Cuerpo de operadores – Cláusula 11.12.1: Cuerpo de transición, Gramática gráfica concreta .....	37
4.157	Error material – Referencia al anexo F – Cláusula 11.12.1: Semántica .....	37
4.158	Aclaración – Sustitución de <action statement> por <action> – Cláusula 11.12.2.2: Unión (Join) .....	37
4.159	Aclaración – <agent body> – Cláusula 11.12.2.2: Unión, Gramática textual concreta .....	37
4.160	Aclaración – <agent body area> – Cláusula 11.12.2.2: Unión, Gramática gráfica concreta .....	37
4.161	Ampliación – Permitir en procedimientos, etc. – Cláusula 11.12.2.3: Parada .....	37
4.162	Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Gramática abstracta .....	37
4.163	Errores materiales – Consistencia entre PR y GR – Cláusula 11.12.2.4: Retorno, Gramáticas concreta .....	38
4.164	Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Gramática gráfica concreta .....	38

	<b>Página</b>
4.165 Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Semántica.....	38
4.166 Error material – Tratamiento de excepción – Cláusula 11.13.1: Tarea, Gramática textual concreta.....	38
4.167 Aclaración y modificación – Cláusula 11.13.1: Tarea .....	38
4.168 Error material – Tratamiento de excepción – Cláusula 11.13.2: Creación, Gramática textual concreta .....	39
4.169 Aclaración – Determinación estática de conjuntos de ejemplares – Cláusula 11.13.2: Creación, Modelo .....	39
4.170 Error material – Inconsistencia de operación/procedimiento – Cláusula 11.13.3: Llamada a procedimiento, Gramática abstracta .....	40
4.171 Error material – Tratamiento de excepción – Cláusula 11.13.3: Llamada a procedimiento, Gramática textual concreta.....	40
4.172 Error material – Inconsistencia entre operación/procedimiento – Cláusula 11.13.3: Llamada a procedimiento, Semántica.....	40
4.173 Error material – Tratamiento de excepción – Cláusula 11.13.4: Salida, Gramática textual concreta.....	41
4.174 Ampliación – Tipos de datos en línea – Cláusula 11.13.4: Salida, Gramática textual concreta.....	41
4.175 Corrección de error – Evaluación de respuestas de decisión – Cláusula 11.13.5: Decisión.....	41
4.176 Aclaración – <terminator/action statement> por <terminator/action> – Cláusula 11.13.5: Decisión, Gramática textual concreta.....	41
4.177 Corrección de error – Evaluación de respuestas de decisión – Cláusula 11.13.5: Decisión.....	41
4.178 Error material – Sintaxis errónea – Cláusula 11.13.5: Decisión, Modelo .....	41
4.179 Ampliación – Permitir stop en procedimientos, etc. – Cláusula 11.14: Lista de enunciados, Gramática textual concreta .....	42
4.180 Error material – Alinear con el anexo F – Cláusula 11.14.1: Enunciado compuesto, Gramática abstracta.....	42
4.181 Error material – Modelo incorrecto – Cláusula 11.14.1: Enunciado compuesto, Modelo.....	42
4.182 Corrección de error – Permitir procedimiento remoto – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados, Gramática textual concreta.....	42
4.183 Aclaración – <action statement> por <action> – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados .....	42
4.184 Error material – tratamiento de excepción – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados, Gramática textual concreta.	42
4.185 Ampliación – Permitir stop en procedimientos, etc. – Cláusula 11.14.2, Acciones de transición y terminadores utilizados como enunciados .....	43
4.186 Aclaración – Cláusula 11.14.4: Enunciado If, Modelo .....	43

	<b>Página</b>
4.187 Error material – Cláusula 11.14.5: Enunciado de decisión .....	43
4.188 Error material – Palabra clave – Cláusula 11.14.6: Enunciado de bucle, Gramática textual concreta.....	44
4.189 Error material – Cláusula 11.14.9: Enunciado excepción, Gramática textual concreta.....	44
4.190 Error material – Tratamiento de excepción – Cláusula 11.15: Temporizador, Gramática textual concreta .....	44
4.191 Error material – Tratamiento de excepción – Cláusula 11.16.2: Excepción activa (On-exception), Gramática textual concreta.....	44
4.192 Aclaración – <action statement> por <action> – Cláusula 11.16.3: Manejo (handle), Modelo .....	44
4.193 Error material – Faltan partes, alineación con el anexo F – Cláusula 12.1: Definiciones de datos, Gramática abstracta.....	44
4.194 Ampliación – Tipos de datos en línea – Cláusula 12.1: Definiciones de datos, Gramática textual concreta .....	45
4.195 Corrección de error – Value/object para anchored sort – Cláusula 12.1: Definiciones de datos, Gramática textual concreta .....	45
4.196 Ampliación – Tipos de datos en línea – Cláusula 12.1: Definiciones de datos, Modelo.....	45
4.197 Aclaración – Cláusula 12.1.1: Definición de tipos de datos.....	46
4.198 Error de sintaxis – <end> – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta .....	46
4.199 Error material – Faltan parámetros de contexto – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta .....	46
4.200 Aclaración – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta.....	46
4.201 Aclaración – Cláusula 12.1.1: Definición de tipos de datos, Gramática gráfica concreta.....	47
4.202 Error material – Supresión de barras verticales en exceso – Cláusula 12.1.2: Definición de interfaces, Gramática textual concreta.....	47
4.203 Error material – Faltan parámetros de contexto – Cláusula 12.1.2: Definición de interfaces, Gramática textual concreta .....	47
4.204 Aclaración – Cláusula 12.1.2: Definición de interfaces, Gramática gráfica concreta .....	47
4.205 Aclaración – Interfaces – Cláusula 12.1.2: Definición de interfaces, Semántica.....	47
4.206 Aclaración – Interfaz implícita – Cláusula 12.1.2: Definición de interfaces, Modelo.....	48
4.207 Error material – Alinear con el anexo F – Cláusula 12.1.4: Operaciones, Gramática abstracta .....	49
4.208 Error material – Ambigüedad de sintaxis – Cláusula 12.1.4: Operaciones, Gramática textual concreta .....	49

	<b>Página</b>
4.209 Error material – Colocación de texto – Cláusula 12.1.5: Any.....	49
4.210 Aclaración – Interfaces implícitas – Cláusula 12.1.6: Pid y géneros de pid, Semántica .....	49
4.211 Error material – Cláusula 12.1.7.3: Tipos de datos elección, Modelo .....	49
4.212 Error material – Géneros facultativos – Cláusula 12.1.8: Comportamiento de los operaciones, Gramática gráfica concreta.....	49
4.213 Error material – Cláusula 12.1.8: Comportamiento de los operaciones, Semántica ..	49
4.214 Ampliación – Range sign – Cláusula 12.1.9.1: Clase de nombre, Gramática textual concreta.....	50
4.215 Error de sintaxis – <end> – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta.....	50
4.216 Error material – Alternativa omitida – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta.....	50
4.217 Modificación – Dificultad para el análisis gramatical (parsing) – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta.....	50
4.218 Error material – Ambigüedad – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta.....	50
4.219 Ampliación – Range sign – Cláusula 12.1.9.5: Condición de intervalo, Gramática textual concreta.....	51
4.220 Error material – Cláusula 12.1.9.6: Definición de sinónimo, Gramática textual concreta.....	51
4.221 Error material – Cláusula 12.2.1: Expresiones, Gramática abstracta.....	51
4.222 Ampliación – Tipos de datos en línea – Cláusula 12.2.1: Expresiones, Gramática textual concreta.....	51
4.223 Error material – Cláusula 12.2.1: Expresiones, Gramática textual concreta.....	51
4.224 Error material – Oración incorrecta – Cláusula 12.2.7: Aplicación de operación, Gramática textual concreta .....	52
4.225 Aclaración – Tratamiento de parámetro – Cláusula 12.2.7: Aplicación de operación, Semántica.....	52
4.226 Corrección de error – Restricción en parámetros – Cláusula 12.2.7: Aplicación de operación, Modelo.....	52
4.227 Corrección de error – Reglas demasiado restrictivas para la inicialización de datos –Cláusula 12.3.1: Definición de variable.....	52
4.228 Error de sintaxis – <end> – Cláusula 12.3.3.2: Inicialización por defecto, Gramática textual concreta .....	53
4.229 Error material – Cláusula 12.3.4.6: Expresión state.....	53
4.230 Error material – Ítems omitidos y duplicados – Cláusula 13.1: Definición facultativa, Gramática textual concreta .....	53
4.231 Aclaración – Cláusula 13.1: Definición facultativa, Gramática textual concreta .....	53
4.232 Aclaración – <action statement> por <action> – Cláusula 13.2: Cadena de transición facultativa, Modelo .....	53

	<b>Página</b>
4.233 Aclaración – <terminator statement> por <terminator> – Cláusula 13.2: Cadena de transición facultativa, Modelo .....	53
4.234 Error material – Palabra clave cuantificación – Cláusula D.2.1: Axiomas, Gramática textual concreta .....	54
4.235 Aclaración – Suprimir la cuantificación implícita – Cláusula D.2.1: Axiomas, Gramática textual concreta .....	54
4.236 Aclaración – Texto informal no utilizado – Cláusula D.2.1: Axiomas, Semántica ...	54
4.237 Aclaración – Suprimir la cuantificación implícita – Cláusula D.2.1: Axiomas, Semántica .....	54
4.238 Aclaración – Letras mayúsculas/minúsculas en los nombres de operador – Cláusula D.3: Lote predeterminado .....	54
4.239 Error material – Falta inherits – Cláusula D.3.13.2: Utilización.....	55
4.240 Error material – Palabras clave con mayúsculas y minúsculas – Apéndice III.....	55
4.241 Aclaraciones – Apéndice III .....	55
4.242 Aclaraciones – Apéndice III .....	55

## Recomendación UIT-T Z.100

### Lenguaje de especificación y descripción

#### CORRIGENDUM 1

### 1 Objetivos y alcance

La finalidad de este corrigendum es registrar los cambios acordados de la Rec. UIT-T Z.100 (1999). Los cambios están agrupados en dos categorías:

- a) Corrección de *errores* y *aclaraciones* (véanse las definiciones en el apéndice II/Z.100).
- b) *Ampliaciones* y *modificaciones* (véanse las definiciones en el apéndice II/Z.100).

Las reglas de mantenimiento en el apéndice II/Z.100 establecen que los *errores* y las *aclaraciones* publicados en la lista maestra de cambios "son efectivas con carácter inmediato". Tales cambios se deben publicar en un corrigendum tan pronto como resulte práctico.

Las *modificaciones* y las *ampliaciones* implican algún cambio en el SDL. La regla en este caso es "a menos que circunstancias especiales requieran que tales cambios se realicen tan pronto como sea posible, dichos cambios no se recomendarán hasta que se revise la Rec. UIT-T Z.100". Las *modificaciones* y las *ampliaciones* incluidas en este corrigendum no se consideran como cambios significativos sino como mejoras útiles de la notación SDL y por lo tanto se recomiendan ahora para su utilización inmediata.

### 2 Terminología

Un *error* es una inconsistencia en una o más de las Recomendaciones UIT-T Z.100 a Z.109.

Una *corrección textual* es un cambio en el texto o diagramas de Recomendaciones que corrige errores materiales o tipográficos.

Un *ítem abierto* es una cuestión identificada pero no resuelta.

Una *deficiencia* es una cuestión identificada en la que las semánticas de SDL no están claramente definidas en las Recomendaciones.

Una *aclaración* es un cambio del texto (o de los diagramas) en una Recomendación que no cambia (intencionalmente) el significado de SDL, sino que tiene por finalidad hacer la Recomendación menos ambigua o más fácil de comprender.

Una *modificación* cambia la semántica de SDL.

Una *ampliación* es una nueva característica que no cambia la semántica de SDL definida en las Recomendaciones aprobadas para SDL.

### 3 Mantenimiento de las Recomendaciones UIT-T Z.100 a Z.109

El apéndice II/Z.100 (1999) documenta el procedimiento que se debe seguir para el mantenimiento de las Recomendaciones Z.100. Este procedimiento requiere que las correcciones de errores, las modificaciones propuestas y las ampliaciones se den a conocer profusamente y que se mantenga una Lista maestra de cambios. Aunque este corrigendum contiene una lista de cambios, la Lista maestra de cambios contiene también otras informaciones no pertinentes a este corrigendum.

## 4 Cambios de la Recomendación Z.100

### 4.1 Error material – Subrayado de <Natural literal name>

Sustitúyanse todas las incidencias de <Natural literal name> por <Natural literal name>.

Éstas se producen en las reglas de sintaxis <number of pages>, <priority name> y <partial regular expression> y en la descripción de <partial regular expression>.

### 4.2 Ampliación – Añádase el <cuerpo de comentario> antes de <left curly bracket>

Añádase [ <comment body> ] antes de cada incidencia de <left curly bracket> en reglas de sintaxis excepto en las reglas léxicas: es decir en 9.4, 11.13.1, 11.14.1, 11.14.5, 12.1, 12.1.1, 12.1.2, 12.1.8 y 12.1.9.4 (en algunas cláusulas, más de una vez).

### 4.3 Error material – <end> – Anexo D

En el anexo D, cada uno de los <data type heading> debe tener un <end>.

### 4.4 Error material – Palabra clave de enunciado de bucle – Cláusula 1.5: Diferencias ...

Reemplazamiento de la palabra clave "for" por palabra clave "loop".

Añádase "loop" a la lista de palabras clave de SDL-2000 que no son palabras clave de SDL-92.

Añádase "for" a la lista de palabras clave de SDL-92 que no son palabras clave de SDL-2000.

### 4.5 Aclaración – Cláusula 5.3.2: Ítems de enumeración titulados, Modelo

Añádase lo siguiente al final de la sección de *Modelo*:

"Las transformaciones pueden ser interdependientes y por lo tanto el orden en que se aplican diversas transformaciones puede determinar la validez y el significado de una especificación SDL. Los detalles precisos del orden de transformación pueden encontrarse en los anexos F1 a F3."

### 4.6 Error material y clarificación – Cláusula 5.4.2: BNF

Cámbiense las cuatro líneas de ejemplo de sintaxis:

```
<diagram in package> ::=  
    | <package diagram>  
    | <package reference area>  
    | <type in agent area>
```

a fin de omitir la barra vertical errónea y red denominar <type in agent area> para que concuerde con una aclaración, como sigue:

```
    <package diagram>  
    | <package reference area>  
    | <entity in agent diagram>
```

Cámbiese "<type in agent area>" por "<entity in agent diagram>" en el párrafo a continuación del ejemplo para que comience como sigue:

"expresa que un <diagram in package> es un <package diagram>, o una <package reference area>, o una <entity in agent diagram>, o una <data type reference area>, ..."

### 4.7 Aclaración – Ejemplo – Cláusula 5.4.3: Metalenguaje para gramática gráfica

Para concordar con una aclaración de la sintaxis actual que cambia <operation graph area> por <operation body area>, cámbiense el Ejemplo y el texto a continuación por:

"Ejemplo:

{ <operation text area>\* <operation body area> } **set**

es un conjunto constituido por cero o más <operation text area>, y una <operation body area>. El metasímbolo **set**" ...

#### 4.8 Error material – Símbolos formados por líneas – Cláusula 5.4.3: Metalenguaje para gramática gráfica

El texto "e implica un flow line symbol (véase 6.5)" no está colocado en el párrafo adecuado. Sustitúyanse los últimos dos párrafos:

"El metasímbolo **is followed by** significa que su argumento de la derecha sigue (tanto lógicamente como en el dibujo) a su argumento de la izquierda.

El metasímbolo **is connected to** significa que su argumento de la derecha está conectado (tanto lógicamente como en el dibujo) a su argumento de la izquierda e implica un flow line symbol (véase 6.5)."

por:

"El metasímbolo **is followed by** significa que su argumento de la derecha sigue (tanto lógicamente como en el dibujo) a su argumento de la izquierda e implica un símbolo de flow line symbol (véase 6.5).

El metasímbolo **is connected to** significa que su argumento de la derecha está conectado (tanto lógicamente como en el dibujo) a su argumento de la izquierda."

#### 4.9 Modificación – Sintaxis de comentario – Cláusula 6.1: Reglas léxicas

La sintaxis de <comment> se cambia a **comment** /# ... #/ para permitir también /# ... #/ antes de "{" como un comentario. Para soportar esto se efectúan las siguientes modificaciones:

Se añade <comment body> como una alternativa de <lexical unit>:

<lexical unit> ::=

```
| <name>
| <quoted operation name>
| <character string>
| <hex string>
| <bit string>
| <note>
| <comment body>
| <composite special>
| <special>
| <keyword>
```

Introdúzcase <number sign> como una alternativa en las siguientes reglas, debido a que se ha suprimido de <other special>:

<note text> ::=

```
{ <general text character>
| <other special>
| <number sign>
| <asterisk>+ <not asterisk or solidus>
| <solidus>
| <apostrophe> }*
```

<not asterisk or solidus> ::=

```
<general text character> | <other special> | <apostrophe> | <number sign>
```

<special> ::=

```
<solidus> | <asterisk> | <number sign> | <other special>
```

Defínase <comment body> mediante las siguientes reglas:

```
<comment body> ::=
    <solidus> <number sign>      <note text> <number sign>+ <solidus>
<comment text> ::=
    {
        <general text character>
        |
        <other special>
        |
        <asterisk>
        |
        <number sign>+ <not number or solidus>
        |
        <solidus>
        |
        <apostrophe> }*
```

```
<not number or solidus> ::=
    <general text character> | <other special> | <apostrophe> | <asterisk>
```

#### 4.10 Ampliación – Señal de intervalo – Cláusula 6.1: Reglas léxicas

Cámbiese la regla <composite special> para incluir la alternativa <range sign> – es decir agréguese una nueva línea después de <result sign>:

```
| <range sign>
```

Añádase después de la regla para <result sign>:

```
<range sign> ::=
    <full stop> <full stop>
```

#### 4.11 Error material – Palabra clave del enunciado de bucle – Cláusula 6.1: Reglas léxicas

La palabra clave **for** debe ser **loop**. Suprímase "**for**" de <keyword> y añádase "**loop**". Reformatéese la disposición de la lista de palabras.

#### 4.12 Error material – Referencia – Cláusula 6.3: Reglas de visibilidad, nombres e identificadores, Gramática textual concreta

En el párrafo:

"Una definición de referencia es una entidad después de la etapa de transformación de <referenced definition> (véase el anexo F)."

Cámbiese la referencia al "anexo F" por "7.3".

#### 4.13 Errores materiales – Contenido de los dibujos – Cláusula 6.6: Partición de dibujos

La palabra "implicit" debe estar subrayada en:

```
<heading area> ::=
    <implicit text symbol> contains <heading>
```

La palabra "drawing" debe estar subrayada en:

```
<kernel heading> ::=
    [<virtuality>] [exported]
    <drawing kind> <drawing qualifier> <drawing name>
```

Inserte la barra vertical que falta al comienzo de la línea:

```
| state [type] | procedure | operator | method
```

La palabra "implicit" debe estar subrayada en:

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

NOTA – Solamente la palabra "Natural" debe estar subrayada en <number of pages> (véase cambio global).

En el texto, la palabra "implicit" debe estar subrayada en "<implicit text symbol>" en el párrafo que comienza:

"El <implicit text symbol> no se muestra, pero está implícito para obtener una clara separación ..."

#### 4.14 Modificación – Cambio de la sintaxis para comentario – Cláusula 6.7: Comentario

Cámbiese la sintaxis por:

```
<comment> ::=
    comment <comment body>
```

#### 4.15 Error material – Gramática para <specification area> – Cláusula 7.1: Marco

Parece que ha habido un problema de redacción con esta cláusula, y la gramática no se completó correctamente. Por tanto, se corrige como sigue:

El comienzo de la regla <specification area>:

```
<specification area> ::=
    <frame symbol> contains
```

se modifica para añadir una llave de apertura como sigue:

```
<specification area> ::=
    <frame symbol> contains {
```

#### 4.16 Error material – Permite señales al entorno para sistemas basados en el tipo – Cláusula 7.1: Marco

Para permitir que las señales al entorno se definan para un sistema basado en el tipo se cambia la sintaxis para permitir lotes (packages) en los sistemas basados en el tipo. Las reglas de sintaxis <textual system specification> y <graphical system specification> se revisan como sigue:

*Gramática textual concreta*

```
<textual system specification> ::=
    <agent definition>
    | {<package use clause>}* <textual typebased agent definition>
```

*Gramática gráfica concreta*

```
<graphical system specification> ::=
    <agent diagram>
    | <graphical typebased agent definition> [ is associated with <package use area> ]
```

y añádanse los siguientes párrafos al final de la sección *Modelo*:

"Una <package use clause> antes de una <textual typebased agent definition> de una <textual system specification> es una sintaxis derivada para una <package use clause> antes del <system heading> en la <system definition> derivada de la <textual typebased agent definition>."

Una <package use area> asociada con una <graphical typebased agent definition> de una <graphical system specification> es una sintaxis derivada para una <package use area> asociada con el <system diagram> derivado de la <graphical typebased agent definition>."

#### 4.17 Aclaración – Tipo ejemplificado potencialmente – Cláusula 7.1: Marco, Semántica

Añádase el siguiente párrafo al final de la *Semántica*:

"Para una *SDL especification* con una *Agent-definition*, un tipo es *ejemplificado potencialmente* si está ejemplificado en la *Agent-definition*, o si está ejemplificado en un tipo ejemplificado potencialmente."

#### 4.18 Aclaración – Sintaxis gráfica concreta para <diagram in package> – Cláusula 7.2: Lote

La alternativa "<type in agent area>" se redenomina "<entity in agent diagram>".

#### 4.19 Error material – Gramática para <specification area> – Cláusula 7.2: Lote, Gramática textual concreta

Cámbiese el texto debajo de la sintaxis de <dependency symbol>:

"... del <package diagram> o <package> o <system specification> para una <package reference area> en una <specification area>, y deben ser consistentes ..."

por:

"... del <package diagram> (o <package> o <system specification> para una <package reference area> en una <specification area>), y deben ser consistentes ..."

#### 4.20 Error material – Unión de lotes – Cláusula 7.2: Lote, Modelo

Suprímase el párrafo:

"Si un lote se menciona en varias <package use clause>s de una <package definition>, ello corresponde a una <package use clause> que selecciona la unión de las definiciones seleccionadas en las <package use clause>."

#### 4.21 Aclaración – Cláusula 8.1.1.1: Tipo de agente, Gramática abstracta

La definición de *Agent-kind*, *Agent-formal-parameter*, *State-machine-definition* se transfiere de la cláusula 9 a esta cláusula.

Añádase *Agent-kind* antes de *Agent-type-identifier*, y *Agent-formal-parameter*, *State-machine-definition* después de *Agent-type-identifier* como sigue:

<i>Agent-kind</i>	=	<b>SYSTEM   BLOCK   PROCESS</b>
<i>Agent-type-identifier</i>	=	<i>Identifier</i>
<i>Agent-formal-parameter</i>	=	<i>Parameter</i>
<i>State-machine-definition</i>	::	<i>State-name</i> <i>Composite-state-type-identifier</i>

#### 4.22 Aclaración – Sintaxis simplificada – Cláusula 8.1.1.1: Tipo de agente, Gramática textual concreta

En la regla <agent type structure> cámbiese "<state partitioning>" por "<state partition>".

Sin embargo, <system type definition>, <block type definition> y <process type definition> se redefinen en términos de <agent structure> en lugar de <agent type structure> y por lo tanto no se requieren las reglas <agent type structure> y <agent type body>. Esto simplifica la sintaxis.

#### 4.23 Aclaración – Sintaxis simplificada – Cláusula 8.1.1.1: Tipo de agente, Gramática gráfica concreta

Cámbiese el nombre de regla "<agent type diagram content>" por "<agent type structure area>".

En forma similar a la gramática textual se suprimen <agent type diagram content>, <type in agent area> y <agent type body area> y <type state machine graph area>, debido a que <system type diagram>, <block type diagram>, y <process type diagram> se definen en términos de <agent structure area>.

#### 4.24 Error material – Definición de conjunto de salida completo – Cláusula 8.1.1.1: Tipo de agente, Semántica

Insértese el siguiente texto al final de *Semántica*:

"El conjunto de salida completo de un tipo de agente es la unión de todas las señales, procedimientos remotos y variables remotas mencionadas, ya sea directamente o como parte de interfaces y listas de señales, en las listas de señales salientes asociadas con las puertas del tipo de agente."

#### 4.25 Aclaración – Modelo de tipo de agente – Cláusula 8.1.1.1: Tipo de agente, Modelo

La descripción de la transformación del modelo estaba incompleta.

Sustitúyase toda la sección *Modelo* por:

"Un tipo de agente con un <agent body> o una <agent body area> es una notación taquigráfica para un tipo de agente que sólo tiene una máquina de estados, pero no agentes contenidos. Esta máquina de estado se obtiene sustituyendo el <agent body> o <agent body area> por una definición de estado compuesto. Esta definición de estado compuesto tiene el mismo nombre que el tipo de agente y su *State-transition-graph* se representa por el <agent body> o la <agent body area>.

Un tipo de agente con:

- una <state partition> con una <composite state reference> o <composite state>, o
- una <state partition area> con una <composite state reference area>, o
- <composite state area>

es una notación taquigráfica para un tipo de agente que tiene una máquina de estados basada en un tipo de estado compuesto implicado virtual. El tipo de estado implicado tiene el cuerpo de la <composite state reference> o <composite state> o <composite state reference area> o <composite state area>. Si el tipo de agente es un subtipo, y si el supertipo tiene una <state partition> o <state partition area>, el tipo de estado implicado es un subtipo que hereda implícitamente el tipo de estado implicado, del supertipo.

Cada tipo implicado tiene una restricción que es él mismo (véase 8.3.1)."

#### 4.26 Error material – Faltan parámetros de contexto – Cláusula 8.1.1.2: Tipo de sistema, Gramática textual concreta

Sustitúyase la regla <system type reference> por:

<system type reference> ::=  
                  **system type** <type reference heading> <type reference properties>

seguido por el nuevo párrafo:

"Un <type reference heading> que es parte de una <system type reference> debe tener un <system type name>."

#### 4.27 Error material – Sintaxis simplificada – Cláusula 8.1.1.2: Tipo de sistema, Gramática textual concreta

Para simplificar la sintaxis, sustitúyase "<agent type structure>" por "<agent structure>" en <system type definition>.

```

<system type definition> ::=
    <package use clause>*
    <system type heading> <end> <agent structure>
endsystem type [ [<qualifier>] <system type name>] <end>

```

#### 4.28 Error material – Puertas/sintaxis omitidas, simplificadas – Cláusula 8.1.1.2: Tipo de sistema, Gramática gráfica concreta

En la regla <system type diagram>, sustitúyase "<agent type diagram content>" por "<agent type structure area>". Sin embargo, en la sintaxis simplificada a continuación <agent type diagram content> se sustituye por <agent structure area>.

Véase también 9.1.

La sintaxis se simplifica para utilizar <agent structure area>. Sustitúyase "<system diagram>" por:

```

<system type diagram> ::=
    <frame symbol> contains {<system type heading> <agent structure area> }
    is connected to {{ <gate on diagram>* }set }
    [ is associated with <package use area> ]

```

#### 4.29 Aclaración – Cláusula 8.1.1.2: Tipo de sistema, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"La <type reference area> que forma parte de una <system type reference area> debe tener un <type reference heading> con un <system type name>."

#### 4.30 Aclaración – Simplificación de sintaxis – Cláusula 8.1.1.3: Tipo de bloque, Gramática textual concreta

Modifíquese la regla <block type definition> para que se utilice "<agent structure>" en lugar de "<agent type structure>".

#### 4.31 Error material – Faltan parámetros de contexto – Cláusula 8.1.1.3: Tipo de bloque, Gramática textual concreta

Sustitúyase la regla <block type reference> por:

```

<block type reference> ::=
    <type preamble>
    block type <type reference heading> <type reference properties>

```

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <block type reference> debe tener un <block type name>."

#### 4.32 Aclaración – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta

En la regla <block type diagram>, sustitúyase "<agent type diagram content>" por "<agent type structure area>". Sin embargo, sustitúyase a continuación "<agent type structure area>" por "<agent structure area>" para simplificar la sintaxis.

#### 4.33 Aclaración – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"El <type reference area> que forma parte de un <block type reference area> debe tener un "<type reference heading> con un <block type name>."

#### 4.34 Corrección de error – Cláusula 8.1.1.3: Tipo de bloque, Gramática gráfica concreta

Suprímase la frase:

"El <gate on diagram> en un <block type diagram> no puede incluir <external channel identifiers>."

#### 4.35 Aclaración – Simplificación de la sintaxis – Cláusula 8.1.1.4: Tipo de proceso, Gramática textual concreta

Cámbiese <process type definition> para que se utilice "<agent structure>" en lugar de "<agent type structure>".

#### 4.36 Error material – Faltan parámetros de contexto – Cláusula 8.1.1.4: Tipo de proceso, Gramática textual concreta

Cámbiese la regla <process type reference> por:

```
<process type reference> ::=  
    <type preamble>  
    process type <type reference heading> <type reference properties>
```

seguido por el nuevo párrafo:

"Un <type reference heading> que es parte de una <process type reference> debe tener un <process type name>."

#### 4.37 Aclaración – Cláusula 8.1.1.4: Tipo de proceso, Gramática gráfica concreta

En la regla <process type diagram> cámbiese "<agent type diagram content>" por "<agent type structure area>". Sin embargo, cámbiese a continuación "<agent type structure area>" por "<agent structure area>" para simplificar la sintaxis.

#### 4.38 Aclaración – Cláusula 8.1.1.4: Tipo de proceso, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"El <type reference area> que forma parte de una <process type reference area> debe tener un "<type reference heading> con un <process type name>."

#### 4.39 Corrección de error – Cláusula 8.1.1.4: Tipo de proceso, Semántica

Suprímase la frase:

"El <gate on diagram> en un <process type diagram> no puede incluir <external channel identifiers>."

#### 4.40 Error material – Alineación con el anexo F – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática abstracta

Suprímase "*Connect-node-set*" en *Composite-state-type-definition* para formar la regla:

```
Composite-state-type-definition    ::    State-type-name  
                                       [ Composite-state-type-identifier ]  
                                       Composite-state-formal-parameter*  
                                       State-entry-point-definition-set  
                                       State-exit-point-definition-set  
                                       Gate-definition-set
```

*Data-type-definition-set*  
*Syntype-definition-set*  
*Exception-definition-set*  
*Composite-state-type-definition-set*  
*Variable-definition-set*  
*Procedure-definition-set*  
 [ *Composite-state-graph* | *State-aggregation-node* ]

Cámbiese ":" por "=" en *Composite-state-type-identifier* para formar la regla:

*Composite-state-type-identifier* = *Identifier*

#### 4.41 Corrección de error – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática abstracta

No hay perjuicio si un tipo de estado compuesto tiene puertas que no se utilizan; por lo tanto, se suprime la siguiente condición estática:

"El *Gate-definition-set* debe estar vacío, a menos que el estado compuesto se utilice como una *State-machine-definition*."

#### 4.42 Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta

Sustitúyase la regla <composite state type definition> por:

```

<composite state type definition> ::=
    {<package use clause>}*
    { <composite state type heading> | <state aggregation type heading> }<end>
    <composite state structure>
    endsubstructure state type [ [ <qualifier> ] <composite state type name> ] <end>
  
```

#### 4.43 Error material – Faltan parámetros de contexto – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta

Cámbiese la regla <composite state type reference> por:

```

<composite state type reference> ::=
    <type preamble>
    state type <type reference heading> <type reference properties>
  
```

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <composite state type reference> debe tener un <composite state type name>."

#### 4.44 Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática textual concreta

Sustitúyase la regla <composite state type diagram> por:

```

<composite state type diagram> ::=
    <frame symbol>
    contains {
        { <composite state type heading> | <state aggregation type heading> }
        <composite state structure area>
        is associated with { <graphical state connection point>* }set
        is connected to { { <gate on diagram>* }set }
        [ is associated with <package use area> ]
  
```

#### 4.45 Aclaración – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"La <type reference area> que forma parte de una <composite state type reference area> debe tener un "<type reference heading> con un <composite state type name>."

#### **4.46 Corrección de error – Cláusula 8.1.1.5: Tipo de estado compuesto, Gramática gráfica concreta**

Transfiérase el párrafo:

"Los <gate on diagram>s en un <process type diagram> tienen que estar fuera del marco del diagrama."

de *Semántica* al final de *Gramática gráfica concreta*.

Suprímase la oración en *Semántica* (que debía haber estado en la *Gramática gráfica concreta*):

"El <gate on diagram> en un <composite state type diagram> no puede incluir <external channel identifiers>."

#### **4.47 Error material – Tipo subrayado – Cláusula 8.1.3.1: Definición de sistema basada en el tipo de sistema, Gramática textual concreta**

Cámbiese "<system type expression>" por "<system type expression>" en la regla <typebased system heading>.

#### **4.48 Error material – Transformación innecesaria – Cláusula 8.1.3.1: Definición de sistema basada en el tipo de sistema**

Cámbiese:

*"Semántica*

Una definición de sistema basada en tipo realiza una *Agent-definition* con un *Agent-kind* **SYSTEM** derivado por transformación a partir de un tipo de sistema.

*Modelo*

Una <textual typebased system definition> o una <graphical typebased system definition> se transforman en una <system definition> que tiene las definiciones del tipo de sistema definidas por <system type expression>."

por:

*"Semántica*

Una definición de sistema basada en tipo (typebased) denotada por una <textual typebased system definition> o una <graphical typebased system definition> define una *Agent-definition* con *Agent-kind* **SYSTEM** que es una ejemplificación del tipo de sistema denotado por la <system type expression>."

#### **4.49 Error material – Transformación innecesaria – Cláusula 8.1.3.2: Definición de bloque basada en tipo de bloque**

Cámbiese:

*"Semántica*

Una definición de bloque basada en tipo define *Agent-definitions* de *Agent-kind* **BLOCK** derivada por transformación a partir de un tipo de bloque.

### Modelo

Una <textual typebased block definition> o una <graphical typebased block definition> se transforma en una <block definition> que tiene las definiciones de tipo de bloque definidas por <block type expression>.

Una <inherited block definition> únicamente puede aparecer en una definición de subtipo. Representa el bloque definido en el supertipo de la definición de subtipo. Pueden especificarse canales adicionales conectados a las puertas del bloque heredado."

por:

"Una <inherited block definition> sólo aparecerá en una definición de subtipo. Representa el bloque definido en el supertipo de la definición de subtipo.

NOTA – Se permite especificar canales adicionales conectados a puertas de un bloque heredado.

### Semántica

Una definición de bloque basada en tipo denotada por una <textual typebased block definition> o una <graphical typebased block definition> define una *Agent-definition* de *Agent-kind* **BLOCK** que es una ejemplificación del tipo de bloque denotado por la <block type expression>".

## 4.50 Error material – Transformación innecesaria – Cláusula 8.1.3.3: Definición de proceso basada en tipo de proceso

Cámbiese:

### "Semántica

Una definición de proceso basada en tipo define una *Agent-definition* con *Agent-kind* **PROCESS** derivada por transformación a partir de un tipo de proceso.

### Modelo

Una <textual typebased process definition> o una <graphical typebased process definition> se transforma en una <process definition> que tiene las definiciones de tipo de proceso definidas por <process type expression>.

Una <inherited process definition> únicamente puede aparecer en una definición de subtipo. Representa el proceso definido en el supertipo de la definición de subtipo. Pueden especificarse canales adicionales conectados a las puertas del proceso heredado."

por:

"Una <inherited process definition> sólo aparecerá en una definición de subtipo. Representa el proceso definido en el supertipo de la definición de subtipo.

NOTA – Se permite especificar canales adicionales conectados a puertas del proceso heredado.

### Semántica

Una definición de proceso basada en tipo (typebased) denotada por una <textual typebased process definition> o una <graphical typebased process definition> define una *Agent-definition* con *Agent-kind* **PROCESS** que es una ejemplificación del tipo de proceso denotado por la <process type expression>."

## 4.51 Error material – Falta exported – Cláusula 8.1.4: Tipo abstracto, Gramática concreta

Suprimanse los dos párrafos:

"Un <operation identifier> no debe representar el *Operation-identifier* para el operador Make definido para un tipo de datos abstracto.

Un <literal identifier> no debe representar un *Literal-identifier* que denota una *Literal-signature* de un tipo de datos abstracto."

#### **4.52 Error material – Sort en propiedad de parámetro de señal – Cláusula 8.1.5: Referencias de tipo, Gramática textual concreta**

Cámbiese la producción para <signal parameter property> por:

```
<signal parameter property> ::=  
    <sort>
```

y sustitúyase el párrafo:

"Una <signal parameter property> se corresponde con una lista de parámetros de señal en una definición de señal. La <sort list> debe corresponder con la <sort list> del <signal definition item> de la correspondiente definición de señal. Tiene que haber una sola <signal parameter property> en las <type reference properties> o <type reference area>."

por:

"Una <signal parameter property> corresponde a un parámetro de señal en una definición de señal. El género (sort) debe corresponder a un <sort> en la <sort list> en <signal definition item> de la correspondiente definición de señal. Las <signal parameter property>s en <type reference properties> deben aparecer en el mismo orden que las propiedades correspondientes en la definición de tipo a que se hace referencia."

#### **4.53 Error material – Falta exported – Cláusula 8.1.5: Referencias de tipo, Gramática gráfica concreta**

Esto se autorizó en SDL-92 y se omitió por error en SDL-2000.

Añádase "**exported as**" en <type reference heading>:

```
<type reference heading> ::=  
    <type preamble>  
    [ exported [ as <remote procedure identifier> ] ]  
    [<qualifier>] <name> [<formal context parameters>]
```

Añádase al párrafo después de esta regla:

"Si **exported** se da en un <type reference heading>, el tipo a que se hace referencia tiene que ser un procedimiento exportado y, si también se da un <remote procedure identifier>, el procedimiento tiene que identificar la misma definición de procedimiento remoto."

#### **4.54 Error material – Constricción de punto extremo demasiado restrictiva – Cláusula 8.1.6: Puerta, Gramática textual concreta**

El siguiente párrafo se ha modificado porque contenía una restricción que ya no se necesita:

"El <identifier> de <textual endpoint constraint> debe denotar una definición de tipo de bloque, tipo de proceso o tipo de estado."

#### **4.55 Error material – Identificador erróneo – Cláusula 8.2.1: Parámetros de contexto de tipo de agente, Gramática textual concreta**

La constricción debe ser un <agent type identifier> en lugar de un <agent identifier>.

#### **4.56 Aclaración – Significado del carácter dos puntos – Cláusula 8.2.2: Parámetros de contexto de agente, Gramática textual concreta**

Añádase el texto "(<colon> <agent type identifier>)" al párrafo de modo que se lea:

"Un parámetro de agente real debe identificar una definición de agente. Su tipo debe ser un subtipo del tipo de agente de constricción (**atleast** <agent type identifier>), sin adición de parámetros formales a los del tipo de constricción, o debe ser el tipo denotado por <agent type identifier> (<colon> <agent type identifier>), o debe ser compatible con la <agent signature> formal."

#### **4.57 Aclaración – Enlace con sintaxis abstracta – Cláusula 8.3.1: Adición de propiedades, Gramática textual concreta**

Añádase el siguiente texto antes del párrafo que comienza por "La sintaxis concreta para especialización de tipos de datos":

"La <type expression> de la <specialization> en:

- a) <agent additional heading> representa el *Agent-type-identifier* de *Agent-type-definition* en 8.1.1.1.
- b) <composite state type heading> o <state aggregation type heading> representa el *Composite-state-type-identifier* de *Composite-state-type-definition* en 8.1.1.5.
- c) <procedure heading> representa el *Procedure-identifier* de *Procedure-definition* en 9.4."

#### **4.58 Error material – Referencia – Cláusula 8.3.3: Transición/conservación virtual, Gramática textual concreta**

Sustitúyase la referencia a "11.2.2 (entrada virtual)" por una referencia a "11.3 (entrada virtual)".

#### **4.59 Error material – Referencia a anexo F – Cláusula 8.3.3: Transición/conservación virtual, Semántica**

Sustitúyase el párrafo:

"La transformación de transiciones y conservaciones virtuales en estado asterisco se desarrolla en el anexo F."

por:

"En el subtipo, es la definición, a partir del subtipo, que define la transición o conservación virtual. Una transición o conservación virtual que no está redefinida en una definición de subtipo tiene la definición dada en la definición de supertipo."

#### **4.60 Error material – Faltan parámetros de contexto – Cláusula 8.4: Asociaciones, Gramática textual concreta**

Sustitúyase la regla <association end> por:

```
<association end> ::=
    [<visibility>] [as <role name>] [size <multiplicity>] [ordered]
    { <agent type reference> | <interface reference> | <data type reference> }
```

Suprímase la regla <linked type identifier>, que no se necesita.

Suprímase "<linked type identifier> del" en el párrafo que comienza por "Si un <association end> está precedido ...".

#### **4.61 Aclaración – Cláusula 9: Agentes, Gramática abstracta**

La regla de sintaxis abstracta para *State-transition-graph* se transfiere a 11.11.1.

Las reglas de sintaxis abstracta para *State-machine-definition*, *Agent-kind* y *Agent-formal-parameter* se transfieren a 8.1.1.1.

#### 4.62 Aclaración – Cláusula 9: Agentes, Gramática textual concreta

En la regla <agent structure> cámbiese "<state partitioning>" por "<state partition>".

Sustitúyase "<state partitioning>" por "<state partition>" las dos veces que aparece en el párrafo a continuación de las reglas.

#### 4.63 Errores materiales – Cláusula 9: Agentes, Gramática textual concreta

Añádase la condición que falta relativa a conexiones. Insértese el siguiente nuevo párrafo antes de la regla <agent instantiation>:

"Una <channel to channel connection> no estará contenida dentro de una <agent type definition>."

En la regla <entity in agent>, insértese "<composite state>" como una alternativa; esto se omitió por error.

#### 4.64 Aclaración – Cláusula 9: Agentes, Gramática textual concreta

Suprímase la regla <state machine graph> eliminando las líneas:

```
<state machine graph>
```

```
<state machine graph>::=
```

#### 4.65 Aclaración – Simplificación – Cláusula 9: Agentes, Gramática textual concreta

En la regla <agent body>, para simplificar la sintaxis de manera que <agent body> se puede utilizar tanto en tipos de agente como en agentes, hágase facultativo la [<on exception>] <start> como sigue:

```
[ [<on exception>] <start> ]
```

Antes del párrafo "La utilización y sintaxis de <valid input signal set> se define en la cláusula 9", insértese el nuevo párrafo:

"En <agent body>, <start> sólo puede omitirse en definiciones de tipo de agente."

#### 4.66 Aclaración – Cómo se describe – Cláusula 9: Agentes, Gramática gráfica concreta

Cámbiese el nombre de regla "<agent diagram content>" por "<agent structure area>" y cámbiese "<type in agent area>" en esta regla por "<entity in agent diagram>".

Redefínase <agent body area> como:

```
<agent body area> ::=
```

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

Esto incorpora la regla <state machine graph area>, la cual se suprime por lo tanto.

Cámbiese el nombre de regla "<type in agent area>" por "<entity in agent diagram>"; dicho nombre de regla debe suprimirse en 8.1.1.1 Tipo de agente e insertarse antes de <interaction area> como:

```
<entity in agent diagram> ::=
```

```
<agent type diagram>  
| <agent type reference area>  
| <composite state area>  
| <composite state type diagram>  
| <composite state type reference area>  
| <procedure diagram>  
| <procedure reference area>  
| <data type reference area>  
| <signal reference area>  
| <association area>
```

En la regla <interaction area>, cámbiese "<state machine area>" por "<state partition area>".

Suprímase la regla denominada "<state machine area>".

En la regla <create line endpoint area>, cámbiese "<state machine area>" por "<state partition area>".

Cámbiese la primera oración después de las reglas por:

"Se permite que una <agent text area> contenga una <agent reference> únicamente si el <agent structure area> directamente circundante contiene una <interaction area>."

Cámbiese "<state machine area>" por "<state partition area>" en la oración que sigue a la NOTA 1 para que se lea:

"La <state partition area> de <interaction area> identifica la máquina de estados (estado compuesto) del agente que puede darse directamente como un gráfico de agente o por referencia a una definición de estado."

Después de este párrafo, insértese un nuevo párrafo:

" <start area> sólo puede omitirse en un diagrama de tipo de agente."

#### **4.67 Error material – Cómo se describe – Cláusula 9: Agentes, Gramática gráfica concreta**

Cámbiese el párrafo que comienza por "Si existe una <agent reference area> ..." como sigue:

"Si existe un <agent reference area> para el agente, las <gate property area> asociadas con la <agent reference area> corresponden a los <gate on diagram> asociados con el <agent diagram>. Ninguna <gate property area> asociada con la <agent reference area> contendrá <signal list item>s no contenidos en los correspondientes <gate on diagram> asociados con el <agent diagram>. Se aplica una regla correspondiente si existe una <agent reference> para el agente."

#### **4.68 Error material – Restricción innecesaria en conexión por puerta – Cláusula 9: Agentes, Semántica**

El siguiente párrafo no es correcto porque puede haber canales implícitos y por lo tanto se suprime. El texto proviene de SDL-92, donde era correcto.

"Si una <agent definition> o una <agent type definition>, utilizada en una <textual typebased agent definition>, contiene <channel definition>s y <textual typebased agent definition>s, entonces, cada puerta de las <agent type definition>s de las <textual typebased agent definition>s contenidas debe conectarse a un canal."

#### **4.69 Aclaración – Condición de parada para los agentes iniciales – Cláusula 9: Agentes, Semántica**

Para aclarar el tiempo de vida de un agente que no tiene máquina de estados, insértese el siguiente párrafo después del párrafo respecto a condiciones de parada que comienza con "Cuando la máquina de estados de un agente ..."

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

#### **4.70 Error material – Conjunto completo de señales de entrada válidas – Cláusula 9: Agentes, Semántica**

Sustitúyase:

- "a) el conjunto de señales de todos los canales o puertas que conducen al conjunto de ejemplares de agentes;

- b) el <valid input signal set>;
- c) las señales de entrada implícitas introducidas como en 10.5, 10.6, 11.4, 11.5 y 11.6; y
- d) "

por:

- "a) el conjunto de señales de todos los canales o puertas que conducen a la máquina de estados del agente;
- b) el <valid input signal set> del agente;
- c) el <valid input signal set> de la máquina de estados del agente;
- d) las señales de entrada implícitas introducidas como en 10.5 y 10.6, y
- e) "

#### 4.71 Error material – Plural incorrecto – Cláusula 9: Agentes, Semántica

Al final de la NOTA 2, sustitúyase "coincidencia de nombres" por "coincidencias de nombre".

#### 4.72 Error material – Definición del conjunto de salida completo – Cláusula 9: Agentes, Semántica

Insértese el siguiente texto al final de *Semántica*:

"El conjunto de salida completo de un conjunto de agentes es el mismo que el conjunto de salida completo del tipo del conjunto de agentes."

#### 4.73 Aclaración – Valor pid de agentes contenidos – Cláusula 9: Agentes, Modelo

Generalícese el penúltimo párrafo de *Modelo* para abarcar los ejemplares iniciales de cualquier agente creado (no solamente del sistema), reemplazándolo así:

"En todos los ejemplares iniciales creados cuando se crea el ejemplo contenido, **parent** se inicializa a Null."

#### 4.74 Error material – Puertas omitidas – Cláusula 9.1: Sistema, Gramática gráfica concreta

Cámbiese la regla <system diagram> por:

```
<system diagram> ::=
    <frame symbol> contains {<system heading> <agent structure area> }
    is connected to { {<gate on diagram>}* <external channel identifiers> }set
    [ is associated with <package use area> ]
```

#### 4.75 Error material – Condición incorrecta – Cláusula 9.1: Sistema, Gramática gráfica concreta

Cámbiese la última oración por:

"Una <system reference area> sólo debe utilizarse como parte de un <specification area>."

#### 4.76 Aclaración – Cláusula 9.2: Bloque, Gramática gráfica concreta

En la regla <block diagram>, cámbiese "<agent diagram content>" por "<agent structure area>", y cámbiese:

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

por:

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

#### 4.77 Aclaración – Cláusula 9.3: Proceso, Gramática gráfica concreta

En la regla <process diagram>, cámbiese "<agent diagram content>" por "<agent structure area>", y cámbiese:

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

por:

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

#### 4.78 Error material – Identificador de procedimiento de nodo/base de arranque – Cláusula 9.4: Procedimiento, Gramática abstracta

En la definición de *Procedure-definition*, sustitúyase "*Procedure-identifier*" por "[*Procedure-identifier*]".

En la definición de *Procedure-graph*, sustitúyase "*Procedure-start-node*" por "[*Procedure-start-node*]".

Al final de la *Gramática abstracta*, añádase el párrafo:

"En una *SDL-specification*, todos los procedimientos potencialmente ejemplificados deben tener un *Procedure-start-node*."

#### 4.79 Error material – Falta exported – Cláusula 9.4: Procedimiento, Gramática textual concreta

Esto estaba autorizado en SDL-92 y se omitió por error en SDL-2000.

Cámbiese la regla <procedure reference> por:

<procedure reference> ::=  
    <type preamble> [ **exported** [ **as** <remote procedure identifier> ] ]  
    **procedure** <procedure identifier> <type reference properties>

Añádase un nuevo párrafo antes de la *Gramática gráfica concreta*:

"Si **exported** se da en una referencia de procedimiento, el procedimiento referenciado tiene que ser un procedimiento exportado y si se da también un <remote procedure identifier> el procedimiento tiene que identificar la misma definición de procedimiento remoto."

#### 4.80 Error material – Faltan parámetros de contexto – Cláusula 9.4: Procedimiento, Gramática textual concreta

Cámbiese la regla <procedure reference> por:

<procedure reference> ::=  
    <type preamble>  
    **procedure** <type reference heading> <type reference properties>

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <procedure reference> debe tener un <procedure name>."

#### 4.81 Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta

En la regla <procedure diagram>, cámbiese "<procedure graph area>" por "<procedure body area>", porque la regla ha sido red denominada (véase más adelante).

#### 4.82 Error material – Falta estado compuesto – Cláusula 9.4: Procedimiento, Gramática gráfica concreta

Cámbiese la regla <procedure area> para permitir diagramas y referencias de tipo de estado compuesto:

```
<procedure area> ::=
    <procedure diagram>
    | <procedure reference area>
    | <composite state type diagram>
    | <composite state type reference area>
```

#### 4.83 Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"La <type reference area> que forma parte de una <procedure reference area> debe tener un "<type reference heading> con un <procedure name>."

#### 4.84 Aclaración – Cláusula 9.4: Procedimiento, Gramática gráfica concreta

Redenomínese la regla <procedure graph area> como "<procedure body area>".

Reemplácese <procedure graph area> por "<procedure body area>" para obtener el siguiente párrafo revisado:

"El <on exception association area> de una <procedure body area> <procedure graph area> identifica al manejador de excepciones asociado a todo el gráfico. El extremo origen no debe estar conectado a ningún símbolo."

#### 4.85 Error material – Cláusula 9.4: Procedimiento, Semántica

En la primera oración de:

"a) Para cada *In-parameter*, se crea una variable local que tiene el *Name* y el *Sort* del *In-parameter*...",

reemplácese el guión en el último "*In-parameter*" de manera que sea un guión normal.

#### 4.86 Error material – Referencia al anexo F – Cláusula 9.4: Procedimiento, Semántica

Sustitúyase el párrafo:

"Un procedimiento externo es un procedimiento cuyo <procedure body> no está incluido en la descripción SDL. Conceptualmente, se supone que a un procedimiento externo se le da un <procedure body> y que se transformará en una <procedure definition> como parte de la transformación de sistema genérica (véase el anexo F)."

por:

"Un procedimiento externo es un procedimiento cuyo <procedure body> no está incluido en la descripción SDL (véase la cláusula 13)."

#### 4.87 Error material – Referencia al anexo F – Cláusula 10.1: Canal, Gramática textual concreta

Sustitúyase el párrafo:

"Si <gate> está especificada, el canal está conectado a esa puerta. La puerta y el canal deben tener al menos un elemento común en sus listas de señales en el mismo sentido. Si no se especifica <gate> alguna se aplican las reglas siguientes:

- a) si el punto extremo de un canal es un agente o máquina de estados y ese agente/estado contiene una <channel to channel connection> para el canal, el canal está conectado a la puerta implícita que introduce la <channel to channel connection>;
- b) si el punto extremo de un canal es un estado, el canal está conectado a la puerta implícita de esa máquina de estados (véase el anexo F),

en cualquier otro caso, el canal introduce una puerta implícita en el agente o servicio mencionado en <channel endpoint>. Esta puerta obtiene la <signal list> de los <channel path>es respectivos como sus correspondientes limitaciones de puerta. El canal está conectado a esa puerta."

por:

"Si <gate> está especificada, el canal está conectado a esa puerta. La puerta y el canal deben tener al menos un elemento común en sus listas de señales en el mismo sentido. Si no se especifica <gate> alguna, se aplica lo siguiente: si el punto extremo de un canal es un agente o máquina de estados y ese agente/estado contiene una <channel to channel connection> para el canal, el canal está conectado a la puerta implícita que introduce la <channel to channel connection>. De lo contrario, el canal introduce una puerta implícita en el agente o estado mencionado en <channel endpoint>. Esta puerta obtiene la <signal list>de los respectivos <channel path> como su constricción de puerta correspondiente. El canal se conecta a esa puerta."

Añádase el siguiente párrafo al final de la sección:

"Si una <channel definition> contiene dos <channel path> se deben satisfacer las siguientes condiciones:

- El <channel endpoint> a continuación de *from* en el primer <channel path> debe ser el mismo que el <channel endpoint> a continuación de *to* en el segundo <channel path>.
- El <channel endpoint> a continuación de *to* en el primer <channel path> debe ser el mismo que el <channel endpoint> a continuación de *from* en el segundo <channel path>".

#### **4.88 Aclaración – Cómo se describió – Cláusula 10.1: Canal, Gramática gráfica concreta**

Cámbiese el párrafo que comienza por "Para cada punta de flecha en el <channel symbol>" por:

"Para cada punta de flecha en el <channel symbol> debe haber como máximo una <signal list area>. Cada <signal list area> tiene que estar suficientemente próxima a una de las puntas de flecha. Esta punta de flecha indica el sentido del trayecto del canal y la lista de señales a la que está asociado."

#### **4.89 Aclaración – Cómo se describió – Cláusula 10.1: Canal, Gramática gráfica concreta**

En la regla <channel definition area> cámbiense ambos ejemplares de "<state machine area>" por "<state partition area>" y sustitúyase la oración a continuación de las reglas de sintaxis por:

"Cuando un <channel symbol> está conectado a una <state partition area>, la <state partition area> denota la máquina de estados del agente que circunda directamente la definición de canal."

#### **4.90 Error material – Referencia al anexo F – Cláusula 10.1: Canal, Modelo**

El texto:

"Los detalles se describen en el anexo F."

fue sustituido primeramente por:

"Los detalles se describen con el modelo para canales implícitos."

Sin embargo, posteriormente el modelo se pasó del proyecto del anexo F a 10.1 y el texto de modelo se pasó del anexo F revisado para obtener el siguiente texto:

### "Modelo

Si se omite el <channel name> en una <channel definition> o <channel definition area>, el canal recibe un nombre de forma implícita e inequívoca.

Un canal en el que los dos puntos extremos son puertas de una <textual typebased agent definition> representa canales individuales desde cada uno de los agentes de dicho conjunto hasta todos los agentes del conjunto, incluyendo el agente de origen. Todo canal bidireccional resultante que conecte un agente del conjunto con ese mismo agente se divide en dos canales unidireccionales.

Si un agente o un tipo de agente contiene puertas explícitas o implícitas que no están conectadas por medio de canales explícitos, se derivan canales implícitos de conformidad con las siguientes tres transformadas, que se deben utilizar después que se aplique la transformada para la creación basada en tipo mencionada en 11.13.2.

- 1) *Transformada 1*: inserción de canales entre conjuntos de ejemplares dentro del agente o el tipo de agente y entre los conjuntos de ejemplares y la máquina de estados del agente;
- 2) *Transformada 2*: inserción de canales desde una puerta en el agente o en el tipo de agente hasta puertas en conjuntos de ejemplares dentro del agente o tipo de agente y hasta puertas en la máquina de estados del agente;
- 3) *Transformada 3*: inserción de canales desde puertas en conjuntos de ejemplares dentro del agente o tipo de agente y desde puertas en la máquina de estados del agente hasta puertas en el agente o tipo de agente.

Estas transformadas se describen en detalle a continuación. Se aplican en el orden dado.

En las transformadas un elemento de la lista de señales (interfaces, señales, procedimientos remotos o variables remotas) concuerda con otro elemento de lista de señales si:

- a) ambos denotan la misma interfaz, señal, procedimiento remoto o variable remota; o
- b) el primero denota una señal o un procedimiento remoto o una variable remota, y el segundo denota una interfaz y la interfaz incluye la señal o el procedimiento remoto o la variable remota; o
- c) ambos denotan interfaces, y el segundo elemento de lista de señales hereda el primer elemento de lista de señales.

*Transformada 1*: inserción de canales implícitos entre entidades dentro de un agente o tipo de agente

- a) si un elemento de la lista de señales salientes asociado con una puerta de un ejemplar en un agente (o tipo de agente) concuerda con un elemento de una lista de señales entrantes asociado con una puerta de otro ejemplar en el mismo agente (o tipo de agente respectivamente); y
- b) si ninguna de estas puertas tiene un canal explícito conectado al mismo,

entonces:

- i) si no existe ningún canal implícito entre las dos puertas, se crea un canal implícito unidireccional desde la puerta por donde el elemento sale hasta la puerta por donde el elemento entra, y este canal no tiene retardo si está dentro de un proceso (o tipo de proceso) y de lo contrario tiene retardo; y
- ii) el elemento se añade a la lista de señales del canal implicado.

*Transformada 2*: inserción de canales implícitos desde las puertas en un agente o tipo de agente

- a) si un elemento de la lista de señales entrantes asociado con una puerta fuera de un agente (o tipo de agente) concuerda con un elemento de una lista de señales entrantes asociado con una puerta de un ejemplar en el agente (o tipo de agente respectivamente); y

- b) si no hay canal explícito dentro del agente (o tipo de agente respectivamente) conectado a la puerta fuera del agente (o tipo de agente respectivamente) y no hay canal explícito conectado a la puerta del ejemplar dentro del agente (o tipo de agente respectivamente)

entonces:

- i) si no existe canal implícito entre las dos puertas, se crea un canal implícito unidireccional desde la puerta fuera del agente (o tipo de agente respectivamente) hasta la puerta del ejemplar dentro del agente (o tipo de agente respectivamente), y este canal no tiene retardo si está dentro de un proceso (o tipo de proceso) y de lo contrario tiene retardo; y
- ii) el elemento se añade a la lista de señales del canal implicado.

*Transformada 3: inserción de canales implícitos desde las puertas en ejemplares*

Se aplica lo siguiente para la inserción de canales implícitos desde las puertas en los conjuntos de ejemplares dentro del agente o tipo de agente hasta las puertas en el agente o tipo de agente;

- a) si un elemento de la lista de señales salientes asociado con una puerta fuera de un agente (o tipo de agente) concuerda con un elemento de una lista de señales salientes asociado con una puerta de un ejemplar en el agente (o tipo de agente respectivamente); y
- b) si no existe un canal explícito conectado a la puerta fuera del agente (o tipo de agente respectivamente) y no existe un canal explícito conectado a la puerta del ejemplar dentro del agente (o tipo de agente respectivamente)

entonces:

- i) si no existe un canal implícito entre las dos puertas en el sentido hacia la puerta fuera del agente (o tipo de agente respectivamente), se crea un canal implícito unidireccional desde la puerta del ejemplar dentro del agente (o tipo de agente respectivamente) hacia la puerta fuera del agente (o tipo de agente respectivamente), y este canal no tiene retardo si está dentro de un proceso (o tipo de proceso) o de lo contrario tiene retardo; y
- ii) el elemento se añade a la lista de señales del canal implicado."

#### **4.91 Error material – Cláusula 10.2: Conexión, Gramática gráfica concreta**

Sustitúyase:

"Gráficamente, el constructivo connect está representado por la conexión gráfica de un <channel symbol> en una <channel definition area> a <external channel identifiers> en una <gate on diagram>."

por:

"Gráficamente, el constructivo connect está representado por la conexión gráfica de un <channel symbol> en un <channel definition area> a un <external channel identifiers> conectado al símbolo de marco circundante."

#### **4.92 Error material – Cláusula 10.2: Conexión, Modelo**

Añádase el siguiente párrafo al final de la sección:

"Cuando un diagrama está directamente contenido dentro de otros diagramas (es decir no está referenciado), se omite cada <external channel identifiers> porque los canales externos conectados al mismo punto en el marco del diagrama desde fuera del diagrama se muestran directamente."

#### **4.93 Error material – Faltan parámetros de contexto – Cláusula 10.3: Señal, Gramática textual concreta**

Cámbiese la regla <procedure reference> por:

<signal reference> ::=  
    <type preamble>  
    **signal** <type reference heading> <type reference properties>

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <signal reference> debe tener un <signal name>."

#### **4.94 Aclaración – Cláusula 10.3: Señal, Gramática gráfica concreta**

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente el párrafo revisado:

"El <type reference area> que forma parte de un <signal reference area> debe tener un "<type reference heading> con un <signal name>."

#### **4.95 Error material – Falta interfaz – Cláusula 10.4: Definición de lista de señales, Gramática textual concreta**

En el párrafo que comienza con:

"Un <signal list item> que es un <identifier> denota un <signal identifier> o <timer identifier> si ..." cámbiese "<timer identifier>" por "<timer identifier> o <interface identifier>."

#### **4.96 Error material – Falta interfaz – Cláusula 10.4: Definición de lista de señales, Gramática textual concreta**

Suprímase el párrafo:

"Sólo está permitido utilizar **this** en <signal list>s que sean parte de las <gate constraint>s (véase 8.1.6)."

El párrafo es erróneo porque la sintaxis no permite "**this**".

#### **4.97 Error material – Falta excepción – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta**

Añádase una "<on exception>" facultativa a "<remote procedure call>" para obtener la regla:

<remote procedure call> ::=  
    **call** <remote procedure call body> [ <on exception> ]

#### **4.98 Error material – Definición del conjunto de salida completo – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta**

En el siguiente párrafo cámbiese "anexo F" por "8.1.1.1 y cláusula 9" (véase cambio por 8.1.1.1):

"Un procedimiento remoto mencionado en una <remote procedure call> debe estar en el conjunto de salida completo (véase el anexo F) de un tipo de agente o conjunto de agentes circundante."

#### **4.99 Error material – Definición del conjunto de salida completo – Cláusula 10.5: Procedimientos remotos, Gramática textual concreta**

No debe permitirse especificar que una definición de procedimiento no tiene retardo.

Suprímase "[ **nodelay** ]" en la sintaxis para <remote procedure definition>.

Suprímase la oración:

"En 10.1 se describe la utilización de **nodelay**".

#### 4.100 Errores materiales – Cambio de renglón suplementario, tabulador suplementario – Cláusula 10.5: Procedimientos remotos, Modelo

Desde el SDL:

```
decision newn = n;  
(true):  
(false): nextstate pWAIT;  
enddecision;  
return;
```

suprímase la línea:

```
return;
```

Suprímase un tabulador al comienzo de la línea con dos tabulaciones después de "... se añade la siguiente:"

```
input pCALL(fpar,n);
```

#### 4.101 Error material – nodelay – Cláusula 10.5: Procedimientos remotos, Modelo

No debe permitirse especificar que una definición de procedimiento remoto no tiene retardo. Suprímase la oración:

"Cuando un procedimiento remoto se transporta en canales explícitos, se ignora la palabra clave **nodelay** de la <remote procedure definition>."

#### 4.102 Error material – Falta excepción – Cláusula 10.6: Variables remotas, Gramática textual concreta

Sustitúyase la regla <export> por las dos reglas:

```
<export> ::=  
                export <export body> [ <on exception> ]  
<export body> ::=  
                ( <variable identifier> { , <variable identifier> }* )
```

#### 4.103 Error material – Referencia al anexo F y nodelay – Cláusula 10.6: Variables remotas

No debe permitirse especificar que una variable remota tiene **nodelay**.

Suprímase "[ **nodelay** ]" en la sintaxis para <remote variable definition> (dos veces).

Suprímase la oración:

"En el anexo F se describe la utilización de **nodelay**."

En el siguiente párrafo, "anexo F" se cambia por "8.1.1.1 y cláusula 9".

Suprímase la oración:

"Una variable remota mencionada en una <import expression> debe estar en el conjunto completo de salidas (véase el anexo F) de un tipo de agente o conjunto de agente circundante."

En *Modelo* suprímase la oración:

"Cuando se transporta una variable remota por canales explícitos, se ignora la palabra clave **nodelay** de la <remote variable definition>."

#### 4.104 Error material – Error de presentación – Cláusula 11.2: Estado, Gramática abstracta

Añádase "*Connect-node-set*" a *State-node* como sigue:



```

<basic state name> ::=
    <state name>
<asterisk state list> ::=
    <asterisk> [ ( <state name> { , <state name> }* ) ]
<composite state item> ::=
    <composite state name> [<actual parameters>]
    | <typebased composite state>
<composite state name> ::=
    <state name>

```

Un <basic state name> es el nombre de un estado que no tiene un <composite state> o <composite state area>, y no está definido en un <typebased composite state>. Un <composite state name> es el nombre de un estado que tiene un <composite state> o <composite state area>, o está definido en un <typebased composite state>.

-----  
e insértense antes de la Gramática gráfica concreta los siguientes dos párrafos (para aplicación de estado compuesto):

"La <connect part> sólo se permite para un <state> con una <state list> que contiene un <composite state item>.

Un <composite state item> o <typebased composite state> contendrá solamente <actual parameters> si está en una <state area> que coincide con una <nextstate area>. En este caso la <state area> debe contener solamente un <composite state name> y, facultativamente, <actual parameters>."

#### 4.107 Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Gramática gráfica concreta

Cámbiese el nombre de regla <basic state area> por "<state area>" y añádase un nuevo ítem asociado alternativo "<connect association area>" al final de la regla para formar la regla (lo cual sustituye también a "<composite state application>") como sigue:

```

<state area> ::=
    <state symbol> contains <state list>
    [ is connected to <on exception association area> ]
    is associated with
    {
        <input association area>
        | <priority input association area>
        | <continuous signal association area>
        | <spontaneous transition association area>
        | <save association area>
        | <connect association area> }*

```

Añádase al final de la sintaxis inmediatamente antes del párrafo que comienza por "Una <state are> representa ...", la regla (de aplicación de estado compuesto):

```

<connect association area> ::=
    <solid association symbol> is connected to <connect area>

```

Añádase, antes de *Semántica* el párrafo:

"<connect association area> sólo se permite para una <state area> con <state list> que contiene un <composite state item>."

#### 4.108 Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado, Semántica

Añádanse al comienzo de *Semántica* los siguientes dos párrafos:

"Un estado representa ya sea una aplicación de estado básico o una aplicación de estado compuesto.

La semántica para la aplicación de estado compuesto se da en 11.11."

Cámbiese el comienzo del siguiente párrafo "Un estado representa ..." por "Un estado básico representa ...".

#### 4.109 Aclaración – Selección de transición – Cláusula 11.2.1 : Estado básico, Semántica

Cámbiese:

"... en el orden siguiente:

a) ..."

por:

"... en los siguientes pasos. Cada vez que se repiten los pasos, el conjunto de señales considerado se actualiza a las señales en el puerto de entrada; de lo contrario se considera el mismo conjunto en cada paso.

a) ..."

y sustitúyase el inciso d) por:

"d) si no se encontró ninguna señal habilitada, estos pasos se repiten tan pronto como las señales en el puerto de entrada difieren del conjunto de señales ya considerado, o si hay un *Input-node* con una *Provided-expression* que pudiese haber cambiado, o *Continuous-expression* que pudiese haber cambiado. Una *Provided-expression* o *Continuous-expression* puede cambiar solamente si contiene una *NOW-expression*, *Timer-active-expression*, *Any-expression*, o *Variable-access* para una variable definida en un proceso circundante que ha cambiado por la asignación en otro ejemplar de agente o en otra partición de estado."

#### 4.110 Aclaración – Presentación de estado básico y de estado compuesto – Cláusula 11.2: Estado básico, Modelo

En el primer párrafo de Modelo, cámbiese "<state name>, se crea una copia ..." por "ítem <state name>, se crea una copia ...".

En el tercer párrafo de Modelo después de "uno para cada <state name>", añádase "y <composite state name>", y después de "aquellos <state name>s" añádase "y <composite state name>s".

Suprímase toda la antigua cláusula 11.2.2, **Aplicación de estado compuesto**.

#### 4.111 Aclaración – <action statement> por <action> – Cláusula 11.3: Entrada, Modelo

Cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

#### 4.112 Error material – Omisión de on-exception – Cláusula 11.5: Señal continua, Gramática abstracta

Cámbiese:

*Continuous-signal* :: *Continuous-expression* [*Priority-name*] *Transition*

por:

*Continuous-signal* :: [*On-exception*]  
*Continuous-expression*  
[*Priority-name*]  
*Transition*

#### 4.113 Aclaración – Selección de transición – Cláusula 11.5: Señal continua, Semántica

Cámbiese el texto de *Semántica* por lo siguiente:

"La *Continuous-expression* se interpreta como parte del estado al que está asociado su *Continuous-signal* (véase 11.2). Si la *Continuous-expression* retorna el valor booleano predefinido verdadero, la señal continua está habilitada.

La señal continua que tiene el valor más bajo para *Priority-name* tiene la prioridad más alta."

#### **4.114 Aclaración – Selección de transición – Cláusula 11.6: Condición habilitadora, Semántica**

Cámbiese el texto de *Semántica* por:

"La *Provided-expression* de un *Input-node* se interpreta como parte del estado al que está asociado este *Input-node* (véase 11.2).

Una señal en el puerto de entrada se habilita si todas las *Provided-expressions* de un *Input-node* retornan el valor booleano predefinido verdadero, o si el *Input-node* no tiene una *Provided-expression*. La *Provided-expression* de una *Spontaneous-transition* puede ser interpretada en cualquier momento mientras el agente se encuentre en el estado."

#### **4.115 Aclaración – Cláusula 11.7: Conservación (save), Gramática abstracta**

En la gramática:

*Save-signalset* :: *Signal-identifier-set*

Cámbiese ":" por "=" para obtener:

*Save-signalset* = *Signal-identifier-set*

#### **4.116 Aclaración – body area en lugar de graph area – Cláusula 11.10: Etiqueta, Gramática textual concreta**

Cámbiese el párrafo después de la sintaxis que comienza con "El término..." por:

"El término "cuerpo" ("body") se utiliza para hacer referencia a un gráfico de máquina de estados, posiblemente después de la transformación desde una <statement list> y después de la transformación desde un tipo. Un cuerpo hace referencia por tanto a <agent body>, <procedure body>, <operation body>, <agent body area>, <procedure body area>, <operation body area>, o <composite state body>."

#### **4.117 Aclaración – <action statement> por <action> – Cláusula 11.10: Etiqueta, Gramática textual concreta**

En el último párrafo de la *Gramática textual concreta* cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

#### **4.118 Aclaración – body area en lugar de graph area – Cláusula 11.10: Etiqueta, Gramática gráfica concreta**

Cámbiese el párrafo después de la sintaxis que comienza por "Un <in-connector area> ..." por:

"Una <in connector area> representa la continuación de un <flow line symbol> desde una <out connector area> correspondiente con el mismo <connector name> en la misma <state machine graph area> o <agent body area> o <procedure body area>."

#### **4.119 Aclaración – <action statement> por <action> – Cláusula 11.10: Etiqueta, Modelo**

Cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

#### 4.120 Aclaración – <terminator statement> por <terminator> – Cláusula 11.10: Etiqueta, Modelo

Cámbiese "<terminator statement>" por "<terminator>" (véase cambio 4.155).

#### 4.121 Error material – Falta "por" – Cláusula 11.11: Máquina de estados y estado compuesto

En la introducción antes de *Gramática abstracta*, en el segundo párrafo cámbiese "y la especificación de un <state>" por "y por la especificación de un <state>".

#### 4.122 Aclaración – Cláusula 11.11: Máquina de estados y estado compuesto, Semántica

En el último párrafo, cámbiese la penúltima oración:

"El procedimiento de salida se invoca después de que se interprete un *Return-node* del *Composite-state-graph* o cuando se interpreta una transición asociada directamente al *State-node*."

por:

"El procedimiento de salida se invoca después de que se interprete un *Return-node* del *Composite-state-graph* y antes de que se interprete una transición asociada directamente al *State-node*, si existen tales transiciones."

#### 4.123 Error material – Falta Modelo – Cláusula 11.11: Máquina de estados y estado compuesto, Modelo

Añádase la sección *Modelo*:

"*Modelo*

Un *Composite-state* tiene un tipo de estado compuesto anónimo implicado que define las propiedades del estado compuesto.

Un *Composite-state* que es una especialización es una notación taquigráfica para definir un tipo de estado compuesto implícito y un estado compuesto typebased de este tipo."

#### 4.124 Error material – Nodo de comienzo facultativo – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática abstracta

Añádase la sintaxis abstracta para *State-transition-graph*, de la cláusula 9.

En la definición de *State-transition-graph*, sustitúyase "*State-start-node*" por "[*State-start-node*]". La nueva sintaxis es por lo tanto:

```
State-transition-graph      ::      [On-exception]  
                               [State-start-node]  
                               State-node-set  
                               Free-action-set  
                               Exception-handler-node-set
```

Al final de la *Gramática abstracta*, añádase el párrafo:

"En una *SDL-specification*, todos los agentes potencialmente ejemplificados deben tener un *State-start-node*."

#### 4.125 Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Sustitúyase la regla <composite state graph> por las dos reglas siguientes:

```

<composite state graph> ::=
    {<package use clause>}*
    <composite state heading> <end> <composite state structure>
    endsubstructure [ [<qualifier>] <composite state name> ] <end>

```

```

<composite state structure> ::=
    substructure
        [<valid input signal set>]
        {<gate in definition>}*
        <state connection points>*
        <entity in composite state>*
        { <composite state body> | <state aggregation body> }

```

Añádase el siguiente párrafo después de la nueva regla <composite state structure>:

"Una <composite state structure> contendrá un <state aggregation body> solamente si está contenida directamente en una <state aggregation> o una <composite state type definition> con un <state aggregation type heading>, de lo contrario contendrá un <composite state body>. Una <composite state structure> que contiene un <state aggregation body> no tendrá un <valid input signal set>."

#### 4.126 Error material – Especialización omitida – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Añádase una <specialization> facultativa a la regla:

```

<composite state heading> ::=
    state [<qualifier>] <composite state name>
    [<agent formal parameters>] [<specialization>]

```

#### 4.127 Error material – Ítems omitidos – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Añádase <procedure reference> como una alternativa a la producción para <entity in composite state> después de <procedure definition>.

Añádase <composite state> como una alternativa a la producción para <entity in composite state> después de <exception definition>.

#### 4.128 Error material – Regla mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Suprímase la regla <composite state reference>, la cual está mejor colocada en 11.11.2, **Agregación de estado**.

#### 4.129 Aclaración – Oración mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Después del párrafo:

"Exactamente uno de los <start>s será no etiquetado. Cada punto de entrada y punto de salida etiquetado adicional tiene que ser definido por <state connection points> correspondientes."

Añádase el siguiente párrafo de 11.11.3 (nueva redacción):

"Un <start> con un <state entry point name> (un start etiquetado) en un <composite state body> hará referencia solamente a <state entry point>s del <composite state graph> que circunda directamente el <composite state body>. Un <return> con un <state exit point> (un return etiquetado) en un <composite state body> hará referencia solamente a <state exit point>s en el <composite state graph> que circunda directamente el <composite state body>."

#### 4.130 Aclaración – Utilización de puerta – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática textual concreta

Añádase la siguiente oración respecto a puertas inmediatamente antes de la *Gramática gráfica concreta*:

"Una <gate> de un <composite state graph> sólo puede ser un <channel endpoint> en el caso de que <composite state graph> sea el <state partitioning> que representa la máquina de estados de un agente."

#### 4.131 Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta

Reemplácese la regla <composite state graph area> por las dos reglas siguientes:

<composite state graph area> ::=

```
<frame symbol> contains
  { <composite state heading> <composite state structure area> }
is associated with {<graphical state connection point>* } set
is connected to { {<gate on diagram> | <external channel identifiers>}* } set
[ is associated with <package use area> ]
```

<composite state structure area> ::=

```
{ <composite state text area>*
  <entity in composite state area>*
  { <composite state body area> | <state aggregation body area> } } set
```

Añádase la siguiente oración después de la nueva regla <composite state structure area>:

"Una <composite state structure area> contendrá una <state aggregation body area> sólo si está directamente contenida en una <state aggregation area> o en un <composite state type diagram> con un <state aggregation type heading>, de lo contrario contendrá una <composite state body area>."

#### 4.132 Error material – Ítems omitidos – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta

Añádase "<exception definition>" como una alternativa en el <text symbol> de la producción para <composite state text area> después de "<procedure reference>".

#### 4.133 Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta

Sustitúyase el nombre de la regla "<type in composite state area>" por "<entity in composite state area>", que es más correcto.

Sustitúyase el párrafo:

"Tiene que haber al menos un <valid input signal set> en las <composite state text area>s de una <composite state graph area>."

por:

"Tiene que haber al menos un <valid input signal set> en las <composite state text area>s de una <composite state graph area> (o la definición de tipo de estado compuesto correspondiente). Un <valid input signal set> no estará contenido en una <composite state text area> de una <state aggregation area> (o la definición de tipo de estado compuesto correspondiente)."

#### 4.134 Aclaración – Oración mal colocada – Cláusula 11.11.1: Gráfico de estado compuesto, Gramática gráfica concreta

Añádanse los siguientes párrafos (después del párrafo antes elaborado):

"A lo sumo una de las <start area>s será no etiquetada. Cada punto de entrada y de salida etiquetado adicional tiene que ser definido por un <state connection points> correspondiente. Cada <start area> etiquetada adicional contendrá un <state entry point name> diferente.

Una <start area> con un <state entry point name> (un comienzo etiquetado) en una <composite state body area> hará referencia sólo a los <state entry point>s de la <composite state graph area> que circunda directamente la <composite state body area>. Una <return area> con un <state exit point> (un retorno etiquetado) en una <composite state body area> hará referencia sólo a <state exit point>s en la <composite state graph area> que circunda directamente la <composite state body area>."

#### **4.135 Aclaración – Cláusula 11.11.1: Gráfico de estado compuesto, Semántica**

Cámbiese el último párrafo de *Semántica* por:

"Es posible especificar un <composite state> que sólo conste de transiciones asociadas con un estado asterisco, sin <start> ni subestado alguno. Estas transiciones pueden ser terminadas por <dash nextstate> o por <return>. Estas transiciones se aplican cuando el agente o procedimiento está en el estado compuesto. El estado siguiente (nextstate) de tal transición terminada por <dash nextstate> es el estado compuesto; sin embargo, la Exit-procedure-definition y entry-procedure-definition del estado compuesto no son llamadas."

#### **4.136 Aclaración – Cláusula 11.11.2: Agregación de estado**

En la segunda oración del primer párrafo, cámbiese "entrelazada en el nivel de transición" por "de transiciones que se alternan" y suprimase la última oración: "Pueden utilizarse agregaciones de estado para dividir el gráfico de un estado".

#### **4.137 Aclaración – Condiciones – Cláusula 11.11.2: Agregación de estado, Gramática abstracta**

La combinación de condiciones en la *Gramática abstracta* y en la *Gramática gráfica concreta* no está clara. Las condiciones deben estar contenidas en la *Gramática abstracta* solamente.

En la *Gramática abstracta* sustitúyase:

"Todos los puntos de entrada y salida del estado contenedor y de las particiones de estados aparecerán en exactamente una *Connection-definition*."

por:

"Para cada *State-partition*, cada uno de los puntos de entrada del estado contenedor aparecerá en exactamente una *Connection-definition*. Para cada *State-partition*, cada uno de los puntos de salida de la *State-partition* aparecerá en exactamente una *Connection-definition*."

#### **4.138 Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática abstracta**

Cámbiese "*State-partition-set*" por "*State-partitions*" en la oración:

"Los conjuntos de señales de entrada de las *State-partition-set* dentro de un estado compuesto tienen que estar disjuntos."

para obtener:

"Los conjuntos de señales de entrada de las *State-partitions* dentro de un estado compuesto tienen que estar disjuntos."

#### **4.139 Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta**

Sustitúyase <state aggregation> por la regla:

```

<state aggregation> ::=
    {<package use clause>}*
    <state aggregation heading> <end> <composite state structure>
    endsubstructure [ [<qualifier>] <composite state name> ] <end>

```

#### 4.140 Error material – Especialización omitida – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta

Añádase una <specialization> facultativa a la regla:

```

<state aggregation heading> ::=
    state aggregation [<qualifier>] <composite state name>
    [<agent formal parameters>][<specialization>]

```

#### 4.141 Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta

En la regla <state aggregation body>, cámbiese "<state partitioning>" por "<state partition>".

Cámbiese el nombre de regla "<state partitioning>" por "<state partition>".

#### 4.142 Error material – Permitir empty body – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta

En la regla <state aggregation body>, cámbiese "+" por "\*" para formar la repetición de cero o más como sigue:

```

<state aggregation body> ::=
    {
        <state partition>
    |
        <state partition connection> }*

```

#### 4.143 Aclaración – Referencia de estado compuesto – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta

<composite state reference> se transfiere de 11.11.1 a esta sección. La sintaxis es:

```

<composite state reference> ::=
    state substructure <composite state name> referenced <end>

```

#### 4.144 Error material – Referencia de estado compuesto – Cláusula 11.11.2: Agregación de estado, Gramática textual concreta

Los identificadores para los puntos de entrada deben ser para estados compuestos y no para particiones de estado. La sintaxis revisada es:

```

<outer entry point> ::=
    <composite state identifier> via <point>

```

```

<inner entry point> ::=
    <composite state identifier> via <point>

```

#### 4.145 Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

Sustitúyase <state aggregation area> por la regla:

```

<state aggregation area> ::=
    <frame symbol> contains {
        <state aggregation heading>
        <composite state structure area>
    } is associated with {<graphical state connection point>* } set
    is connected to { {<gate on diagram> | <external channel identifiers>}* } set
    [ is associated with <package use area> ]

```

#### 4.146 Error material – Permitir empty body – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

En la regla <state aggregation body area>, cámbiese "<connection definition area>" por "<state partition connection area>" y cámbiese "+" por "\*" para formar la repetición de cero o más como sigue:

```
<state aggregation body area> ::=
    { { <state partition area> | <state partition connection area>* } set
```

(La regla <connection definition area> se redenomina <state partition connection area> – véase más adelante.)

#### 4.147 Aclaración – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

En la regla <state partition area>, cámbiese "<state partition reference area>" por "<composite state reference area>".

Cámbiese el nombre de regla "<state partition reference area>" por "<composite state reference area>".

#### 4.148 Error material – Identificador – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

Modifíquese la regla <inherited state partition definition> tal como se refiere en <composite state identifier> en lugar de <state partition identifier>.

#### 4.149 Error material – Sintaxis de puntos de conexión – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

Sustitúyanse las reglas:

```
<connection definition area> ::=
    <solid association symbol> is connected to <graphical point>
```

```
<graphical point> ::=
    <graphical state connection point>
    | { <state entry points> | <state exit points> } is associated with <state partition
area>
    | <state connection point symbol> is connected to <frame symbol>
```

por lo siguiente donde la regla <connection definition area> se redenomina "<state partition connection area>":

```
<state partition connection area> ::=
    <solid association symbol> is connected to
    [ <outer graphical point> <inner graphical point> ]
```

```
<outer graphical point> ::=
    { <state entry points> | <state exit points> } is associated with <frame symbol>
```

```
<inner graphical point> ::=
    { <state entry points> | <state exit points> } is associated with <state partition
area>
```

#### 4.150 Error material – <gate>s omitidas – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

Se omitió permitir puertas en la sintaxis gráfica de estados (aunque estaban autorizadas en la forma textual correspondiente). Las reglas <composite state reference area> y <graphical type based state partition definition> se actualizan como sigue, y el texto a continuación se añade después de las reglas:

<composite state reference area> ::=  
    <state symbol> **contains** { <state name> {<gate>\*}**set** }

<graphical typebased state partition definition> ::=  
    <state symbol> **contains** { <typebased state partition heading> {<gate>\*}**set** }

Las <gate>s contenidas en <state symbol>s se colocan cerca del borde de los símbolos y se asocian con el punto de conexión a canales.

Una <gate> se permite en un <state symbol> de una <graphical typebased state partition definition> o <composite state reference area> de una <state partition area> sólo si la <state partition area> representa la máquina de estados de un agente o tipo de agente.

#### 4.151 Aclaración – Condiciones – Cláusula 11.11.2: Agregación de estado, Gramática gráfica concreta

La combinación de condiciones en la *Gramática abstracta* y la *Gramática gráfica concreta* no está clara. Las condiciones deben estar contenidas en la *Gramática abstracta* solamente.

En la *Gramática gráfica concreta*, suprimáanse las condiciones:

"El mismo punto de salida de estado de una partición de estado no podrá estar conectado a más de un punto de salida de estado del estado circundante.

Un punto de entrada de estado del estado circundante no podrá estar conectado a dos puntos de entrada de estados diferentes, de la misma partición de estado."

#### 4.152 Aclaraciones – Cláusula 11.11.2: Agregación de estado, Semántica

En el primer párrafo, tercera oración, cámbiese "creación de las *State-partition-set* contenidas" por "creación de cada una de las *State-partition* contenidas" y cámbiese "*State-partition-set* y sus conexiones" por "*State-partition* y sus conexiones".

#### 4.153 Aclaración – Oración mal colocada – Cláusula 11.11.3: Punto de conexión de estado, Semántica

Suprimáse la siguiente oración de 11.11.3 (las adiciones a 11.11.1, y el texto de 11.12.2.4 tienen el mismo efecto):

"Un <composite state> sólo puede referirse a sus propios <state entry point>s en <start>s etiquetados, y a sus propios <state exit point>s en <return>s etiquetados".

#### 4.154 Ampliación – Lista de conexión (connect list) más flexible – Cláusula 11.11.4: Connect

Para permitir una sintaxis más flexible se modifican <connect list> y <asterisk connect list> y se añade <state exit point list> como sigue:

```
<connect list> ::=  
    <state exit point list>  
    | <asterisk connect list>  
<state exit point list> ::=  
    { <state exit point name> | default } { , { <state exit point name> | default } } *  
<asterisk connect list> ::=  
    <asterisk> [ ( <state exit point list> ) ]
```

La sección de *Modelo* se sustituye por:

"**default** en una <state exit point list> representa un <return> no etiquetado.

Cuando la <connect list> de una determinada <connect part> contiene más de un <state exit point name>, se crea una copia de la <connect part> para cada uno de esos <state exit point name>. A continuación la <connect part> es sustituida por estas copias.

Una <connect list> que contiene una <asterisk connect list> se transforma en una lista de <state exit point>s, uno para cada <state exit point> del <composite state> en cuestión (incluyendo el <return> no etiquetado) excepto los escritos en paréntesis después del <asterisk>. La lista de <state exit point>s se transforma entonces tal como se ha descrito anteriormente."

#### 4.155 Error material – Tratamiento de excepción – Cláusula 11.12.1: Cuerpo de transición, Gramática textual concreta

En la *Gramática textual concreta* redenominense "<terminator statement>" y "<action statement>" por "<terminator>" y "<action>" respectivamente, y suprimanse las reglas <action 1>, <action 2>, <terminator 1> y <terminator 2> para obtener las siguientes reglas de sintaxis:

```

<transition> ::=
    { <transition string> [ <terminator> ] }
    |
    <terminator>

<transition string> ::=
    { <action> }+

<action> ::=
    [ <label> ]
    {
        | <task>
        | <output>
        | <create request>
        | <decision>
        | <set>
        | <reset>
        | <export>
        | <procedure call>
        | <remote procedure call>
        | <transition option> } <end>

<terminator > ::=
    [ <label> ]
    {
        | <return>
        | <raise>
        | <nextstate>
        | <join>
        | <stop>}end>

```

y sustitúyase "<terminator statement>" por "<terminator>" en el texto después de las reglas de sintaxis.

En la *Gramática gráfica concreta*, cámbiense las reglas como sigue:

```

<transition area> ::=
    [ <transition string area> is followed by ]
    <terminator area>

<terminator area> ::=
    | <state area>
    | <nextstate area>
    | <decision area>
    | <stop symbol>
    | <merge area>
    | <out connector area>
    | <return area>
    | <transition option area>
    | <raise area>

```

```

<transition string area> ::=
    <action area>
    [ is followed by <transition string area> ]

<action area> ::=
    <task area>
    | <output area>
    | <create request area>
    | <procedure call area>
    | <remote procedure call area>

```

#### 4.156 Error material – Cuerpo de operadores – Cláusula 11.12.1: Cuerpo de transición, Gramática gráfica concreta

Añádase al final de la sección el párrafo:

"Una <transition area> en una <operation body area> no contendrá una <state area> o una <nextstate area>."

#### 4.157 Error material – Referencia al anexo F – Cláusula 11.12.1: Semántica

En el párrafo que comienza por:

"Una transición en un proceso de un bloque puede ..."

sustitúyase "por las reglas del anexo F" con "por las reglas dadas en las secciones *Modelo* de esta Recomendación".

#### 4.158 Aclaración – Sustitución de <action statement> por <action> – Cláusula 11.12.2.2: Unión (Join)

En la primera oración, cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

#### 4.159 Aclaración – <agent body> – Cláusula 11.12.2.2: Unión, Gramática textual concreta

De acuerdo con la aclaración en lo referente a suprimir "type" con respecto a "bodies", cámbiese "<agent type body>" por "<agent body> en una definición de tipo" en la oración después de la sintaxis.

#### 4.160 Aclaración – <agent body area> – Cláusula 11.12.2.2: Unión, Gramática gráfica concreta

De acuerdo con las aclaraciones relativas a la PR y otras aclaraciones, cámbiese "<state machine graph area>" por "<agent body area>" (dos veces) en la primera oración después de la sintaxis.

#### 4.161 Ampliación – Permitir en procedimientos, etc. – Cláusula 11.12.2.3: Parada

Véanse también 11.14 y 11.14.2. Permitir parada en los procedimientos etc.

En la *Gramática abstracta*, suprimase la condición "Un *Stop-node* no podrá estar contenido en un *Procedure-graph*".

En la *Semántica*, añádase al final:

"La interpretación de un *Stop-node* en un *Procedure-graph* o *State-transition-graph* provoca la parada del agente que interpreta ese *Procedure-graph*. La interpretación del procedimiento, operación, enunciado compuesto, o estado compuesto termina y la parada se propaga hacia afuera al llamante y se trata como si se hubiese interpretado un *Stop-node* en lugar de la llamada de procedimiento, aplicación de operación, invocación del enunciado compuesto, o transición al estado compuesto. La terminación se propaga hacia afuera hasta que se alcanza el agente contenedor."

#### 4.162 Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Gramática abstracta

Añádase al final de la *Gramática abstracta*:

"La *Expression* de un *Value-return-node* debe ser compatible en género con el género (sort) del *Result* del *Procedure* circundante."

#### 4.163 Errores materiales – Consistencia entre PR y GR – Cláusula 11.12.2.4: Retorno, Gramáticas concreta

Cámbiese <return> en la *Gramática textual concreta* por:

```
<return> ::=
                return [<return body>] [ <end> <on exception> ]
```

Cámbiese <return area> en la *Gramática gráfica concreta* por:

```
<return area> ::=
                <return symbol>
                [ is connected to <on exception association area> ]
                [ is associated with <return body> ]
```

y añádase:

```
<return body> ::=
                <expression>
                | via <state exit point>
```

#### 4.164 Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Gramática gráfica concreta

Añádase al final de la *Gramática gráfica concreta*:

"La <expression> en <return> o <return area> no se omitirá si el alcance circundante es un operador o método con un <operation result> o un procedimiento que retorna un valor con un <procedure result> sin un <variable name>."

NOTA – Si se omite la <expression> en un operador o método con un <operation result> o un procedimiento que retorna un valor con un <procedure result> denominado, el modelo en 9.4 añade la variable de resultado de procedimiento como la <expression>."

#### 4.165 Error material – Género de retorno – Cláusula 11.12.2.4: Retorno, Semántica

Cámbiese:

"c) Si se interpreta un *Value-return-node*, el resultado de *Expression* retorna al contexto llamante."

por:

"c) Si se interpreta un *Value-return-node*, el resultado de *Expression* se interpreta de la misma manera que una *Expression* asignada a una variable con el género (sort) del resultado (véase 12.3.3), pero sin que el resultado esté asociado con una variable o tenga lugar una comprobación de la gama (o intervalo); a continuación, el resultado de objeto o de valor se retorna al contexto llamante."

#### 4.166 Error material – Tratamiento de excepción – Cláusula 11.13.1: Tarea, Gramática textual concreta

Cámbiese la regla <task> por:

```
<task> ::=
                task <textual task body> [ <end> <on exception> ]
```

#### 4.167 Aclaración y modificación – Cláusula 11.13.1: Tarea

En la *Gramática textual concreta*, cámbiese la última alternativa de <textual task body> por:

| [ <comment body> ] <left curly bracket> <statement list> <right curly bracket>

En la *Gramática gráfica concreta*, cámbiese <task area> por:

<task area> ::=

| <task symbol> **contains** <graphical task body>  
| [ is connected to <on exception association area> ]  
| <macro symbol> **contains** { <macro name> [<macro call body>] }

y añádase al final de las reglas:

"<macro symbol> ::=



En la *Gramática gráfica concreta*, insértese al final, antes de la *Semántica*:

"Una <task area> o <task> que contiene una sola <assignment> representa una *Assignment* o *Assignment-attempt*.

Una <task area> o <task> que contiene un solo <set> representa un *Set-node*.

Una <task area> o <task> que contiene un solo <reset> representa un *Reset-node*.

Una <task area> o <task> que contiene cualquier otra <statement list> representa un *Compound-node*. El *Connector-name* se representa por un nombre anónimo de nueva creación. El *Variable-definition-set* se representa por la lista de todas las <variable definition>s en <statement list>. La *Transition* se representa por la transformada de <statements> en <statement list>, o por la transformada de <statements> en <statement list> seguida por un *Break-node* con *Connector-name*, si la <statement list> no es de terminación."

En la *Semántica*, insértese después del primer párrafo un nuevo párrafo:

"La interpretación de un *Compound-node* se da en 11.14.1. La interpretación de *Assignment* y *Assignment-attempt* se da en 12.3.3. La interpretación de *Set-node* y *Reset-node* se da en 11.15."

En el *Modelo*, sustitúyanse todos los párrafos existentes, incluyendo la Nota, por:

"Si la <statement list> de <textual task body> o <graphical task body> está vacía, la <task> o <task area>, respectivamente, se suprime. Todo ítem sintáctico que conduzca a tal <task> o <task area> vacías conducirá por tanto directamente al ítem que sigue a la <task> o <task area>, respectivamente.

Un <macro symbol> que contiene <macro name> [<macro call body>] se transforma en un <task symbol> que contiene **macro** <macro name> [<macro call body>] <end>."

#### 4.168 Error material – Tratamiento de excepción – Cláusula 11.13.2: Creación, Gramática textual concreta

Cámbiese la regla <create request> por:

<create request> ::=

**create** <create body> [ <end> <on exception> ]

#### 4.169 Aclaración – Determinación estática de conjuntos de ejemplares – Cláusula 11.13.2: Creación, Modelo

El inciso c) de *Modelo* implicaba que se pueden crear dinámicamente conjuntos de ejemplares, en cuyo caso los canales implícitos al conjunto de ejemplares no se podrían derivar de la misma manera

que los conjuntos de ejemplares estáticos. Por el contrario, se ha determinado estáticamente que se puede crear un ejemplar y, por lo tanto, si no se da explícitamente un conjunto de ejemplares, se considera implícitamente un conjunto de ejemplares (sin ejemplares iniciales). El texto del inciso c) en *Modelo* se modifica como sigue:

- "c) si no hay un conjunto de ejemplares del tipo agente indicado en el agente contenedor, entonces:
- i) se considera que en el agente contenedor está implícito un conjunto de ejemplares del tipo dado con un nombre único; y
  - ii) el `<agent type identifier>` en la `<create request>` es una sintaxis derivada para este conjunto de ejemplares implícito."

#### **4.170 Error material – Inconsistencia de operación/procedimiento – Cláusula 11.13.3: Llamada a procedimiento, Gramática abstracta**

Cámbiese:

"Cada *Expression* correspondiente por posición a un *Inout-parameter* o *Out-parameter* tiene que ser un *Variable-identifier* compatible en género con el género identificado por el *Sort-reference-identifier* del *Procedure-formal-parameter*."

por:

"Cada *Expression* correspondiente por posición a un *Inout-parameter* o *Out-parameter* tiene que ser un *Variable-identifier* con el mismo *Sort-reference-identifier* que el *Procedure-formal-parameter*."

#### **4.171 Error material – Tratamiento de excepción – Cláusula 11.13.3: Llamada a procedimiento, Gramática textual concreta**

Cámbiese la regla `<procedure call>` por:

`<procedure call> ::=`  
`call <procedure call body> [ <end> <on exception> ]`

#### **4.172 Error material – Inconsistencia entre operación/procedimiento – Cláusula 11.13.3: Llamada a procedimiento, Semántica**

La sección *Semántica* se debe sustituir por:

"La interpretación de un *Call-node* o *Value-returning-call-node* de procedimiento interpreta las expresiones de parámetros reales en el orden en que se presentaron. Si no se plantean excepciones por la interpretación del parámetro, la interpretación se transfiere a la definición de procedimiento referenciada por el *Procedure-identifier*, y se interpreta ese gráfico de procedimiento (la explicación se da en 9.4).

Si se omite una `<expression>` en `<actual parameters>` el parámetro formal correspondiente no tiene ítem de datos asociado, es decir, es "undefined".

Si un género (sort) de argumento del *Call-node* o *Value-returning-call-node* para un *In-parameter* o *Inout-parameter* del procedimiento es un syntype, se aplica al resultado de la *Expression* la comprobación de intervalo definida en 12.1.9.5. Si la comprobación de intervalo es el valor booleano predefinido Falso en el momento de la interpretación, entonces se plantea la excepción predefinida *OutOfRangeException* (véase D.3.16) en lugar de interpretar ulteriores parámetros reales o la definición de procedimiento.

Si no se plantea *OutOfRangeException*, la interpretación de la transición que contiene un *Call-node* continúa cuando se termina la interpretación del procedimiento llamado.



hasta que se identifica una *Range-condition* que contiene la *Decision-question*, o hasta que esta determinación requiera la interpretación de una aplicación de operación que plantea una excepción, o se elija un *Informal-text*."

#### 4.178 Error material – Sintaxis errónea – Cláusula 11.13.5: Decisión, Modelo

Cámbiese la especificación del syntype definido anónimamente como sigue:

```
syntype data_type_N =  
  <<package Predefined>>Integer constants 1:N  
endsyntype;
```

#### 4.179 Ampliación – Permitir stop en procedimientos, etc. – Cláusula 11.14: Lista de enunciados, Gramática textual concreta

Véanse también 11.12.2.3 y 11.14.2. Permitir stop en procedimientos, etc.

Añádase una alternativa <stop statement> a <terminating statement>.

#### 4.180 Error material – Alinear con el anexo F – Cláusula 11.14.1: Enunciado compuesto, Gramática abstracta

Añádase un *Exception-handler-node* facultativo a *Compound-node*:

```
Compound-node          :: Connector-name  
                        Variable-definition-set  
                        [ Exception-handler-node ]  
                        Init-graph-node*  
                        Transition  
                        Step-graph-node*
```

#### 4.181 Error material – Modelo incorrecto – Cláusula 11.14.1: Enunciado compuesto, Modelo

El *Modelo* era inconsistente con la semántica directa para <compound statement>.

Suprímase la nota:

"NOTA – La <statement list> no vacía transformada se convierte en una lista de <action statement>s y <terminator statement>s separados por caracteres punto y coma y se termina por punto y coma, por lo que puede tratarse como una <transition>."

y cámbiese la oración:

"Si la <statement list> está vacía, el resultado de su transformación es el texto vacío."

por la siguiente nota:

"NOTA – Si la <statement list> está vacía se representa por un *Break-node* como se explica en la *Gramática gráfica concreta*."

#### 4.182 Corrección de error – Permitir procedimiento remoto – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados, Gramática textual concreta

Las llamadas a procedimiento remoto se han omitido por error. Cámbiese la regla <call statement> por:

```
<call statement> ::=  
  call { <procedure call body> | <remote procedure call body> }
```

#### 4.183 Aclaración – <action statement> por <action> – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados

En la primera oración, cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

#### 4.184 Error material – tratamiento de excepción – Cláusula 11.14.2: Acciones de transición y terminadores utilizados como enunciados, Gramática textual concreta

Cámbiense las siguientes reglas como se indica a continuación:

```
<algorithm action statement> ::=
    output <output body> <end>
    |
    create <create body> <end>
    |
    set <set body> <end>
    |
    reset <reset body> <end>
    |
    export <export body> <end>
<return statement> ::=
    return <return body> <end>
<raise statement> ::=
    raise <raise body> <end>
```

#### 4.185 Ampliación – Permitir stop en procedimientos, etc. – Cláusula 11.14.2, Acciones de transición y terminadores utilizados como enunciados

Véanse también 11.12.2.3 y 11.14. Permitir stop en procedimientos, etc.

En la *Gramática textual concreta*, añádase una producción para <stop statement>:

```
<stop statement> ::=
    stop <end>
```

En el *Modelo*, cámbiense el último párrafo que comienza con "La transformación de ..." por:

"La transformada de un <algorithm action statement>, un <return statement>, un <stop statement>, y un <raise statement> se consigue dejando fuera el <end> final."

#### 4.186 Aclaración – Cláusula 11.14.4: Enunciado If, Modelo

Simplificación de la utilización de *Modelo* para <decision statement>.

Cámbiense:

"El <if statement> es equivalente al siguiente <action statement> que implica una <decision>:

```
decision <Boolean expression> ;
    ( true ) : task { <consequence statement>-transform };
    ( false ) : task { <alternative statement>-transform };
enddecision ;
```

La transformación de <alternative statement> sólo se inserta si existe <alternative statement>."

por:

"El <if statement> es equivalente al siguiente <decision statement>:

```
decision ( <Boolean expression> ) {
    ( true ) : <consequence statement>
    ( false ) : <alternative statement>
}
```

Si no existe <alternative statement>, se inserta un <empty statement> en su lugar."

#### 4.187 Error material – Cláusula 11.14.5: Enunciado de decisión

La alternativa **any** se omitió por error. La sintaxis debe ser:

```
<decision statement> ::=
    decision ( { <expression> | any } ) [ <comment body> ] <left curly bracket>
    <decision statement body>
    <right curly bracket>
```

**4.188 Error material – Palabra clave – Cláusula 11.14.6: Enunciado de bucle, Gramática textual concreta**

Véanse también 5.1, 6.1 y D.2.1.

La palabra clave debe ser **loop**.

Sustitúyase "for" por "loop" en <loop statement>.

**4.189 Error material – Cláusula 11.14.9: Enunciado excepción, Gramática textual concreta**

Sustitúyase "*Local-scope-node*" por "*Compound-node*". (Referencia errónea)

**4.190 Error material – Tratamiento de excepción – Cláusula 11.15: Temporizador, Gramática textual concreta**

Sustitúyanse las reglas <reset> y <set> por las cuatro reglas siguientes:

```
<reset> ::=
                reset <reset body> [ <end> <on exception> ]
<reset body> ::=
                ( <reset clause> { , <reset clause> }* )
<set> ::=
                set <set body> [ <end> <on exception> ]
<set body> ::=
                <set clause> { , <set clause> }*
```

**4.191 Error material – Tratamiento de excepción – Cláusula 11.16.2: Excepción activa (On-exception), Gramática textual concreta**

Suprímase "<end>" de la regla "<on exception>" para obtener:

```
<on exception> ::=
                onexception <exception handler name>
```

**4.192 Aclaración – <action statement> por <action> – Cláusula 11.16.3: Manejo (handle), Modelo**

En el penúltimo párrafo, cámbiese "<action statement>" por "<action>" (véase cambio 4.155).

**4.193 Error material – Faltan partes, alineación con el anexo F – Cláusula 12.1: Definiciones de datos, Gramática abstracta**

Cámbiense las siguientes reglas añadiendo "*Data-type-definition-set*", "*Syntype-type-definition-set*", y "*Exception-definition-set*", para obtener:

```
Value-data-type-definition      ::      Sort
                                     Data-type-identifier
                                     Literal-signature-set
                                     Static-operation-signature-set
                                     Dynamic-operation-signature-set
                                     Data-type-definition-set
                                     Syntype-definition-set
                                     Exception-definition-set

Object-data-type-definition     ::      Sort
                                     Data-type-identifier
                                     Literal-signature-set
                                     Static-operation-signature-set
                                     Data-type-definition-set
                                     Syntype-definition-set
```



#### 4.196 Ampliación – Tipos de datos en línea – Cláusula 12.1: Definiciones de datos, Modelo

Añádanse los siguientes párrafos al final de la sección:

"Un <sort> que es un <basic sort> con una <range condition> es una sintaxis concreta derivada para un <syntype> de una <syntype definition> implicada que tiene un nombre anónimo. Esta <syntype definition> denominada anónimamente se define con sus elementos restringidos por la <range condition>, si el <basic sort> se ha construido utilizando el constructor de tipo de datos literal; de lo contrario, la <range condition> forma parte de una <size constraint>.

Una <inline data type definition> es una sintaxis concreta derivada para un <basic sort> de una <data type definition> implicada que tiene un nombre anónimo. Esta <data type definition> denominada anónimamente se deriva de la <inline data type definition> insertando **type** y el nombre anónimo después de **value** u **object** en la <inline data type definition>. Cada <inline data type definition> define una <data type definition> implicada diferente.

Una <inline syntype definition> es una sintaxis concreta derivada para un <basic sort> de una <syntype definition> implicada que tiene un nombre anónimo. Esta <syntype definition> denominada anónimamente se deriva de la <inline syntype definition> insertando el nombre anónimo y <equals sign> después del syntype en la <inline syntype definition>."

#### 4.197 Aclaración – Cláusula 12.1.1: Definición de tipos de datos

Para aclarar el significado de la construcción de tipo de datos con respecto a valores literales, insértese en 12.1.1 inmediatamente después del encabezamiento:

"Una definición de tipo de datos tiene un cuerpo que generalmente contiene un constructor de tipo de datos y una indicación de si el tipo de datos es un tipo de datos **value** u **object**.

El constructor de tipo de datos define cómo construir conjuntos de valores [(valores estructurados, valores literales y valores de elección (choice values)]. Si la definición de tipo de datos es un tipo **value**, estos valores son los elementos del género (sort). Si la definición de tipo de datos es un tipo **object**, estos valores son lo referenciado por los elementos del género."

#### 4.198 Error de sintaxis – <end> – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta

Véanse también 12.1.9.4 y 12.3.3.2.

Cámbiese la regla <data type definition body> por:

```
<data type definition body> ::=
    {<entity in data type>}* [<data type constructor>] <operations>
    [<default initialization> <end> ]
```

#### 4.199 Error material – Faltan parámetros de contexto – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta

Cámbiese la regla <data type reference> por:

```
<data type reference> ::=
    <type preamble>
    { value | object } type <type reference heading> <type reference properties>
```

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <data type reference> debe tener un <data type name>."

#### 4.200 Aclaración – Cláusula 12.1.1: Definición de tipos de datos, Gramática textual concreta

Añádase al final:

"Para cada <operation signature> de <operation signatures> habrá una y solamente una definición correspondiente (<operation definition> o <textual operation reference> o <external operation definition>) en las <operation definitions> de las <operations>."

#### 4.201 Aclaración – Cláusula 12.1.1: Definición de tipos de datos, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para obtener el siguiente párrafo revisado:

"La <type reference area> que forma parte de una <data type reference area> debe tener un "<type reference heading> con un <data type name>."

#### 4.202 Error material – Supresión de barras verticales en exceso – Cláusula 12.1.2: Definición de interfaces, Gramática textual concreta

Suprímense las barras verticales para alternativas antes de <virtuality> en:

```
|   {<package use clause>}*  
|   [<virtuality>] <interface heading>  
|   [<interface specialization>] <end>  
|   {<package use clause>}*  
|   [<virtuality>] <interface heading>
```

para obtener el texto:

```
|   {<package use clause>}*  
|   [<virtuality>] <interface heading>  
|   [<interface specialization>] <end>  
|   {<package use clause>}*  
|   [<virtuality>] <interface heading>
```

#### 4.203 Error material – Faltan parámetros de contexto – Cláusula 12.1.2: Definición de interfaces, Gramática textual concreta

Cámbiese la regla <interface reference> por:

```
<interface reference> ::=  
    [<virtuality>]  
    interface <type reference heading> <type reference properties>
```

seguido por el nuevo párrafo:

"Un <type reference heading> que forma parte de una <interface reference> debe tener un <interface name>."

#### 4.204 Aclaración – Cláusula 12.1.2: Definición de interfaces, Gramática gráfica concreta

Sustitúyase "<graphical type reference heading> que contenga" por "<type reference heading> con" para formar el párrafo revisado:

"La <type reference area> que forma parte de una <interface reference area> debe tener un "<type reference heading> con un <interface name>."

#### 4.205 Aclaración – Interfaces – Cláusula 12.1.2: Definición de interfaces, Semántica

Para definir herencia de interfaz, el contenido de una interfaz debe estar definido:

En la *Semántica*, insértese después del primer párrafo el siguiente nuevo párrafo:

"El contenido de una interfaz es el conjunto de todas las señales, procedimientos remotos y variables remotas que están definidos en una <entity in interface> de la interfaz o referenciados en la <interface use list> o incluidos en la interfaz por especialización (que es la parametrización de herencia o contexto)."

Sin embargo, la definición de interfaces en 12.1.2 no debería tener *Semántica* ya que no tiene *Gramática abstracta*. Por lo tanto, sustitúyase el encabezamiento "*Semántica*" por "*Modelo*" y suprimase el encabezamiento "*Modelo*".

#### **4.206 Aclaración – Interfaz implícita – Cláusula 12.1.2: Definición de interfaces, Modelo**

Véase también 12.1.6.

El modelo no funcionó correctamente. Sustitúyase el texto de *Modelo* por:

"Las interfaces están implícitamente definidas por definiciones de agente y de tipo de agente y por definiciones de máquinas de estado de agente y de tipo de agente. La interfaz definida implícitamente para un agente o un tipo de agente tiene el mismo nombre y está definida en la misma unidad de ámbito que el agente o el tipo de agente que la definió. La interfaz definida implícitamente para una máquina de estados tiene el mismo nombre que el agente o el tipo de agente contenedor pero está definida en la misma unidad de ámbito que la máquina de estados que la definió, es decir, dentro del agente o tipo de agente.

NOTA 1 – Como todo agente y tipo de agente tienen una interfaz implícitamente definida con el mismo nombre, cualquier interfaz definida explícitamente debe tener un nombre diferente del nombre de cada agente y tipo de agente definidos en el mismo ámbito; de lo contrario, habrá coincidencia de nombres.

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.

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.

La interfaz definida por un agente basado en el tipo contiene en su <interface specialization> la interfaz definida por su tipo.

NOTA 2 – Para evitar textos recargados, se sigue el convenio de utilizar la frase "el género pid del agente A" en lugar de "el género pid definido por la interfaz definida implícitamente por el agente A" cuando no sea probable que esto cree confusión."

#### 4.207 Error material – Alinear con el anexo F – Cláusula 12.1.4: Operaciones, Gramática abstracta

Añádase *Identifier* a:

*Operation-signature* :: *Operation-name*  
*Formal-argument\**  
[*Result*]  
*Identifier*

y añádase el siguiente párrafo después de las reglas de gramática:

"El *Identifier* en una signature de operador es un identificador anónimo para el procedimiento anónimo que corresponde a la operación."

#### 4.208 Error material – Ambigüedad de sintaxis – Cláusula 12.1.4: Operaciones, Gramática textual concreta

Cámbiese la regla <operation preamble> por:

<operation preamble> ::=  
[ <virtuality> [<visibility>] | <visibility> [<virtuality>] ]

#### 4.209 Error material – Colocación de texto – Cláusula 12.1.5: Any

Suprímase las últimas dos líneas antes del encabezamiento de *Gramática textual concreta*.

El resto del texto de la sección explicativa no normativa, excepto el primer párrafo, se debe colocar al comienzo de la sección *Semántica*.

#### 4.210 Aclaración – Interfaces implícitas – Cláusula 12.1.6: Pid y géneros de pid, Semántica

Véase también 12.1.2.

Cámbiese:

"b) si la señal está contenida en una puerta de entrada conectada a la máquina de estados del agente que ha definido implícitamente la interfaz; o

c) si la verificación de compatibilidad ..."

por:

"b) si la verificación de compatibilidad ...".

#### 4.211 Error material – Cláusula 12.1.7.3: Tipos de datos elección, Modelo

[*Esta corrección no concierne al texto español.*]

#### 4.212 Error material – Géneros facultativos – Cláusula 12.1.8: Comportamiento de los operaciones, Gramática gráfica concreta

Suprímase el párrafo:

"Para cada <operation diagram> debe existir una <operation signature> en la misma unidad de ámbito ... Igualmente, <sort> puede omitirse en <operation result>."

#### 4.213 Error material – Cláusula 12.1.8: Comportamiento de los operaciones, Semántica

Cámbiese:

"Una <external operation definition> es un operador o método cuyo comportamiento no está incluido en la descripción SDL. Conceptualmente, se asume que una <external operation definition>

recibe un comportamiento y se transforma en una <operation definition> como parte de la transformación de sistema genérica (véase el anexo F)."

por:

"Una <external operation definition> es un operador o método cuyo comportamiento no está incluido en la descripción SDL (véase la cláusula 13)."

#### 4.214 Ampliación – Range sign – Cláusula 12.1.9.1: Clase de nombre, Gramática textual concreta

Véanse también 6.1 y 12.1.9.5.

Cámbiese la regla <regular interval> por:

```
<regular interval> ::=
    <character string> { <colon> | <range sign> } <character string>
```

#### 4.215 Error de sintaxis – <end> – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta

Véanse también 12.1.1 y 12.3.3.2.

Cámbiese la primera alternativa en la producción para <syntype definition> y suprimase la barra vertical antes de <type preamble> como una alternativa para formar la regla:

```
<syntype definition> ::=
    {<package use clause>}*
    syntype <syntype name> <equals sign> <parent sort identifier>
        { { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end>
          <syntype closing>
          | [<syntype closing>] } <end>
    | {<package use clause>}*
      <type preamble> <data type heading> [<data type specialization>]
```

#### 4.216 Error material – Alternativa omitida – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta

Añádase la alternativa omitida por error en <syntype definition> como sigue:

```
| syntype <syntype name> <equals sign> <parent sort identifier>
  <left curly bracket>
    [ { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end> ]
  <right curly bracket>
```

#### 4.217 Modificación – Dificultad para el análisis gramatical (parsing) – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta

Cámbiese la segunda producción para <syntype definition> como sigue:

```
| {<package use clause>}*
  <type preamble> <data type heading> [<data type specialization>]
  {<end> <data type definition body> <constraint> <end> <data type closing> <end>
  | <left curly bracket> <data type definition body> <constraint> <end>
  <right curly bracket> }
```

#### 4.218 Error material – Ambigüedad – Cláusula 12.1.9.4: Sintipos, Gramática textual concreta

Añádase el siguiente párrafo después de la sintaxis al final de *Gramática textual concreta*:

"Si una <constraint> puede interpretarse como perteneciente a la <default initialization> o a la <syntype definition> se debe considerar que forma parte de la <default initialization>."

#### 4.219 Ampliación – Range sign – Cláusula 12.1.9.5: Condición de intervalo, Gramática textual concreta

Véanse también 6.1 y 12.1.9.1.

Cámbiese la regla <closed range> por:

```
<closed range> ::=
                <constant> { <colon> | <range sign> } <constant>
```

#### 4.220 Error material – Cláusula 12.1.9.6: Definición de sinónimo, Gramática textual concreta

Cámbiese:

"Un <external synonym definition item> define un <synonym> cuyo resultado no está definido en una especificación. Conceptualmente, se supone que un <external synonym definition item> recibe un resultado y se transforma en un <internal synonym definition item> como parte de la transformación de sistema genérica (véase anexo F)."

por:

"Un <external synonym definition item> define un <synonym> cuyo resultado no está definido en una especificación (véase la cláusula 13)."

#### 4.221 Error material – Cláusula 12.2.1: Expresiones, Gramática abstracta

Añádase "*State-expression*" como una alternativa a "*Active-expression*".

#### 4.222 Ampliación – Tipos de datos en línea – Cláusula 12.2.1: Expresiones, Gramática textual concreta

Sustitúyase "<expression>" por las siguientes dos reglas:

```
<expression> ::=
                <expression0>
                | <range check expression>
<expression0> ::=
                <operand>
                | <create expression>
                | <value returning procedure call>
```

Suprímase "<range check expression>" en <operand 2> para obtener la regla:

```
<operand2> ::=
                <operand3>
                | <operand2> { <greater than sign>
                              | <greater than or equals sign>
                              | <less than sign>
                              | <less than or equals sign>
                              | in } <operand3>
                | <equality expression>
```

Cámbiese "<constant expression>" por:

```
<constant expression> ::=
                <constant expression0>
```

Cámbiese "Una <expression> que ..." por "Una <expression0> que ..."

#### 4.223 Error material – Cláusula 12.2.1: Expresiones, Gramática textual concreta

En el primer párrafo, cámbiese:

"Una <expression> que no contenga ningún <active primary>, una <create expression>, o un <value returning procedure call> con un <remote procedure call body> es una <constant expression>. Una <constant expression> representa una *Constant-expression* en la sintaxis abstracta. ..."

para obtener lo siguiente con "constant" subrayado:

"Una <expression> que no contiene ningún <active primary>, <create expression>, o <value returning procedure call> es una <constant expression>. Una <constant expression> representa una *Constant-expression* en la sintaxis abstracta. ..."

Subráyese "constant" en el segundo párrafo como sigue:

"Una <expression> que no es una <constant expression> representa una *Active-expression*."

#### **4.224 Error material – Oración incorrecta – Cláusula 12.2.7: Aplicación de operación, Gramática textual concreta**

Suprímase el párrafo "Si se utiliza **this** <operation identifier> tiene que denotar un operador o método circundante".

#### **4.225 Aclaración – Tratamiento de parámetro – Cláusula 12.2.7: Aplicación de operación, Semántica**

Suprímase el párrafo que comienza por "Si se omite una <expression> en <actual parameters>, ...".

#### **4.226 Corrección de error – Restricción en parámetros – Cláusula 12.2.7: Aplicación de operación, Modelo**

Para permitir que se aplique un método a un primario que no es una variable, es necesario añadir lo siguiente a la sección *Modelo*:

"Si el <primary> de una <method application> no es una variable o **this**, hay una asignación implícita del <primary> a una variable implícita con el género (sort) del primer parámetro de la operación (que es el género del método). La asignación se coloca antes de la acción en la que se produce la <method application>. La variable implícita sustituye al <primary> en la <method application>."

#### **4.227 Corrección de error – Reglas demasiado restrictivas para la inicialización de datos – Cláusula 12.3.1: Definición de variable**

Las reglas para la inicialización eran demasiado estrictas e impedían una inicialización razonable de objetos.

En la condición de *Gramática abstracta*:

"Si está presente la *Constant-expression* ..."

se sustituye por:

"Si la *Constant-expression* está presente, debe ser una de las siguientes:

- 1) el mismo género que el *Sort-reference-identifier* denotado, o
- 2) si el género denotado es un OS de género de objeto, el género denotado por *value OS*, o
- 3) si el género denotado es un VS de género de valor, el género denotado por *object OS*".

Cámbiese el primer párrafo de la *Semántica*, por:

"Cuando se crea una variable y existe la *Constant-expression*, la variable se asocia con:

- 1) el resultado de la *Constant-expression*, si el género (sort) de la variable y la *Constant-expression* son los mismos;

- 2) un objeto (distinto de cualquier otro objeto) que hace referencia a la *Constant-expression*, si la variable tiene un género de objeto y la *Constant-expression* tiene un género de valor;
- 3) el valor referenciado por la *Constant-expression*, si la variable tiene un género de valor y la *Constant-expression* tiene un género de objeto.

De lo contrario, si no se aplica ninguna *Constant-expression*, la variable no tiene un ítem de datos asociado: esto es, la variable es "undefined".

#### 4.228 Error de sintaxis – <end> – Cláusula 12.3.3.2: Inicialización por defecto, Gramática textual concreta

Véanse también 12.1.9.4 y 12.1.1.

Cámbiese la regla para <default initialization> como sigue:

```
<default initialization> ::=
    default [<virtuality>] [<constant expression>]
```

#### 4.229 Error material – Cláusula 12.3.4.6: Expresión state

Véase *Gramática abstracta* en 12.2.1, Expresiones.

Sustitúyase el contenido de toda la sección por el texto siguiente:

*Gramática abstracta*

```
State-expression      ::      ()"
```

*Gramática textual concreta*

```
<state expression> ::=
    state
```

*Semántica*

Una expresión de estado (state expression) denota el literal (Charstring) que contiene el nombre correcto exacto del último estado a que ha pasado la unidad de ámbito circundante más cercana. Si no existe tal estado, <state expression> denota la cadena vacía (").

#### 4.230 Error material – Ítems omitidos y duplicados – Cláusula 13.1: Definición facultativa, Gramática textual concreta

Añádase "<agent reference>", "<signal reference>", e "<interface reference>" como alternativas a la producción para <select definition>.

Suprímase la alternativa para "<textual typebased state partition definition>" de <select definition>.

#### 4.231 Aclaración – Cláusula 13.1: Definición facultativa, Gramática textual concreta

En la regla <select definition>, cámbiese "<state partitioning>" por "<state partition>".

#### 4.232 Aclaración – <action statement> por <action> – Cláusula 13.2: Cadena de transición facultativa, Modelo

Cámbiese "<action statement>" por "<action>" en los incisos a) y b) (véase cambio 4.155).

#### 4.233 Aclaración – <terminator statement> por <terminator> – Cláusula 13.2: Cadena de transición facultativa, Modelo

Cámbiese "<terminator statement>" por "<terminator>" en el inciso a) (véase cambio 4.155).

#### **4.234 Error material – Palabra clave cuantificación – Cláusula D.2.1: Axiomas, Gramática textual concreta**

Véanse también 5.1, 6.1 y D.2.1.

Añádase la siguiente nota después de la regla <quantification>:

"NOTA – **for** se considera una palabra clave SDL para los fines de este anexo".

#### **4.235 Aclaración – Suprimir la cuantificación implícita – Cláusula D.2.1: Axiomas, Gramática textual concreta**

Sustitúyase:

- "c) un <value identifier> si existe una definición de dicho nombre en una <quantification> de <quantified equations> que circunda el <term>, que debe tener un género adecuado para el contexto; en caso contrario
- d) un <value identifier> que tiene una ecuación cuantificada implícita para la <unquantified equation>.

Dos o más ocurrencias del mismo identificador de valor no limitado dentro de una <unquantified equation>, (o si la <unquantified equation> está contenida en una <conditional equation>, dentro de la <conditional equation>), entrañan una <quantification>.

Dentro de una <unquantified equation> (o en el caso de que la <unquantified equation> esté contenida en una <conditional equation>, dentro de la <conditional equation>), tiene que haber exactamente un género para cada identificador de valor implícitamente cuantificado, que sea consistente con todos los usos."

por:

- "c) un <value identifier> si existe una definición de dicho nombre en una <quantification> de <quantified equations> que circunda el <term>, que debe tener un género adecuado para el contexto."

#### **4.236 Aclaración – Texto informal no utilizado – Cláusula D.2.1: Axiomas, Semántica**

Suprímase el párrafo:

"Si uno o más axiomas cualesquiera contienen texto informal, la interpretación de expresiones no está formalmente definida ... no se ha dado una especificación formal completa en SDL."

#### **4.237 Aclaración – Suprimir la cuantificación implícita – Cláusula D.2.1: Axiomas, Semántica**

Suprímense las oraciones:

"El género de las cuantificaciones implícitas es el género requerido por el contexto (o los contextos) de la ocurrencia del identificador no vinculado. Si los contextos de un identificador de valor que tiene una cuantificación implícita permiten géneros diferentes, el identificador está vinculado a un género que es consistente con todos sus usos en la ecuación."

#### **4.238 Aclaración – Letras mayúsculas/minúsculas en los nombres de operador – Cláusula D.3: Lote predefinido**

Los nombres de operador "Make", "Extract", "Modify" se deben modificar de manera que se escriban como se indica más adelante, en lo que respecta al uso de mayúsculas/minúsculas en todo el anexo:

es decir, la letra inicial debe ser mayúscula y no minúscula.

#### 4.239 Error material – Falta inherits – Cláusula D.3.13.2: Utilización

Insértese el **inherits** que falta en:

```
value type Boolset inherits Bag< Boolean > endvalue type Boolset;
```

#### 4.240 Error material – Palabras clave con mayúsculas y minúsculas – Apéndice III

En el apéndice III cámbiese:

"a) Sustituir todas las palabras clave por las correspondiente <lowercase keyword> (o <uppercase keyword>);"

para reemplazar "<lowercase keyword>" por "lowercase <keyword>" y "<uppercase keyword>" por "uppercase <keyword>" con el fin de obtener el siguiente texto:

"a) Sustituir todas las palabras clave por la correspondiente <keyword> en minúsculas (o <keyword> en mayúsculas);"

y modifíquese el inciso d) para obtener el siguiente texto:

"d) Si <name>s entran en conflicto con <keyword>s en minúsculas, sustitúyase el primer carácter por el mismo carácter en mayúscula."

#### 4.241 Aclaraciones – Apéndice III

En el inciso 1 b), cámbiese "<national>" por "nacionales".

#### 4.242 Aclaraciones – Apéndice III

Sustitúyase "la de las transiciones duplicadas" por "una de las transiciones duplicadas".

## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	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	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
<b>Serie Z</b>	<b>Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación</b>