



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

CCITT

COMITÉ CONSULTIVO
INTERNACIONAL
TELEGRÁFICO Y TELEFÓNICO

Z.100 Anexo A

(11/1988)

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

Lenguaje de especificación y descripción funcionales
(LED)

Criterios para la utilización de técnicas de descripción
formal (TDF)

GLOSARIO LED

Reedición de la Recomendación Z.100 Anexo A del
CCITT publicada en el Libro Azul, Fascículo X.1 (1988)

NOTAS

- 1 La Recomendación Z.100 Anexo A del CCITT se publicó en el fascículo X.1 del Libro Azul. Este fichero es un extracto del Libro Azul. Aunque la presentación y disposición del texto son ligeramente diferentes de la versión del Libro Azul, el contenido del fichero es idéntico a la citada versión y los derechos de autor siguen siendo los mismos (véase a continuación).
- 2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

ANEXO A
(a la Recomendación Z.100)

Glosario LED

La Recomendación Z.100 contiene las definiciones formales de terminología LED. El glosario LED se ha compilado para facilitar a los nuevos usuarios del LED la lectura de la Recomendación y sus anexos, dándoles una breve definición de cada término y una referencia a la sección de la Recomendación en que éste aparece. Las definiciones contenidas en el glosario pueden ser versiones resumidas o expresadas con otras palabras de las definiciones formales, y podrían por tanto ser incompletas.

Los términos escritos en cursiva figuran en el glosario. Si una frase en cursiva, por ejemplo *identificador de procedimiento*, no figura en el glosario, puede ser la concatenación de dos términos, en este caso, el término *identificador* seguido del término *(de) procedimiento*. Las palabras en cursiva que no figuren en el glosario pueden ser palabras derivadas de un término del glosario. Por ejemplo, *exportado* es el participio pasado de *exportar*.

Excepto cuando un término es sinónimo de otro, tras la definición del término hay una referencia principal al empleo del término en las Recomendaciones de la serie Z.100. Estas referencias se indican entre corchetes [] después de las definiciones. Por ejemplo [3.2], indica que la referencia principal está en el § 3.2.

acceder

E: *access*

F: *accès*

Acceder es la operación aplicada a una *variable* que da el último valor asignado a ésta. Si se accede a una *variable* que tiene un *valor indefinido*, se produce un *error*.

acceso de entrada

E: *inlet*

F: *accès entrant*

Un *acceso de entrada* representa una línea (por ejemplo, un *canal* o una *línea de flujo*) que entra en una *llamada a macro LED/GR*. [4.2.3]

acceso de salida

E: *outlet*

F: *accès sortant*

Un *acceso de salida* representa una línea (por ejemplo, un *canal* o una *línea de flujo*) que sale de un *diagrama de macro*. [4.2.2]

acción

E: *action*

F: *action*

Una *acción* es una operación que se ejecuta dentro de una *cadena de transición*, p. ej., una *tarea*, *salida*, *decisión*, *petición de crear* o *llamada a procedimiento*. [2.7]

álgebra inicial

E: *initial algebra*

F: *algèbre initiale*

Un *álgebra inicial* es el formalismo para definir *tipos abstractos de datos*. [5.3]

área ; zona

E: *area*

F: *zone*

Un *área* es una región bidimensional en la *sintaxis gráfica concreta*. Las *áreas* a menudo corresponden a nodos en la *sintaxis abstracta* y usualmente contienen *sintaxis textual común*. En *diagramas de interacción*, las *áreas* pueden ser conectadas por *canales* o *rutas de señales*. En *diagramas de control de flujo* las *áreas* pueden ser conectadas por *líneas de flujo*.

área de asociación

E: association area

F: zone d'association

Un *área de asociación* es una conexión entre *áreas* en un *diagrama de interacción* por medio de un *símbolo de asociación*. Hay cinco *áreas de asociación*: *área de asociación de subestructura de canal*, *área de asociación de entrada*, *área de asociación de entrada con prioridad*, *área de asociación de señal continua* y *área de asociación de conservación*. [2.6.3, 3.2.3, 4.10.2, 4.11]

área de bloque

E: block area

F: zone de bloc

El *área de bloque* es la definición de un *bloque* o una referencia a un *bloque* en un *diagrama de interacción*. [2.4.2]

área de cadena de transición

E: transition string area

F: zone de chaîne de transition

El *área de cadena de transición* es la representación *LED/GR* de una *cadena de transición*. [2.6.7.1]

área de condición habilitante (o habilitadora)

E: enabling condition area

F: zone de condition de validation

El *área de condición habilitante* es la representación *LED/GR* de una *condición habilitante*. [4.12]

área de conservación

E: save area

F: zone de mise en réserve

El *área de conservación* es la representación *LED/GR* de una *conservación*. [2.6.5]

área de decisión

E: decision area

F: zone de décision

Un *área de decisión* es la representación *LED/GR* de una *decisión*. [2.7.5]

área de definición de canal

E: channel definition area

F: zone de définition de canal

El *área de definición de canal* es la definición de un *canal LED/GR*. [2.5.1]

área de entrada

E: input area

F: zone d'entrée

Un *área de entrada* es la representación *LED/GR* de una *entrada*. [2.6.4]

área de estado

E: state area

F: zone d'état

Un *área de estado* es la representación *LED/GR* de uno o más *estados*. [2.6.3]

área de fusión

E: merge area

F: zone de fusion

Un *área de fusión* es aquélla en que una *línea de flujo* se conecta con otra. [2.6.7.2.2]

área de línea de crear

E: create line area

F: zone de ligne de création

El *área de línea de crear* en un *diagrama de bloque* conecta el *área de proceso* del *proceso creador* (*PROGENITOR*) con el *área de proceso* del *proceso creado* (*VÁSTAGO*). [2.4.3]

área de lista de señales

E: signal list area

F: zone de liste de signaux

El *área de lista de señales* en un *diagrama de interacción* representa una *lista de señales* asociada a un *canal* o a una *ruta de señales*. [2.5.5]

área de llamada a (de) procedimiento

E: procedure call area

F: zone d'appel de procédure

El *área de llamada a procedimiento* es la representación *LED/PR* de una *llamada a procedimiento*. [2.7.3]

área de opción general

E: general option area

F: zone d'option générale

El *área de opción general* es la representación *LED/GR* de una *opción*. [4.3.3]

área de proceso

E: process area

F: zone de processus

Un *área de proceso* en *LED/GR* es la representación de un *proceso* o una referencia a un *proceso* en un *diagrama de interacción*. [2.4.3]

área de salida

E: output area

F: zone de sortie

El *área de salida* en un *diagrama de flujo de control* representa el concepto *LED/GR* de una *salida*. [2.7.4]

área de servicio

E: service area

F: zone de service

Un *área de servicio* es un *diagrama de servicio* o una referencia a un *servicio*. [4.10.1]

área de tarea

E: task area

F: zone de tâche

Un *área de tarea* es la representación *LED/GR* de una *tarea*. [2.7.1]

área de transición

E: transition area

F: zone de transition

Un *área de transición* es la representación *LED/GR* de una *transición*. [2.6.7.1]

arranque

E: start

F: départ

El *arranque* en un *proceso* es interpretado antes de cualquier *estado* o *acción*. El *arranque* inicializa el *proceso* reemplazando sus *parámetros formales* por los *parámetros efectivos* especificados en el crear. [2.6.2]

asignar

E: assign

F: affectation

Asignar es la operación aplicada a una *variable* que asocia a la *variable* un *valor* que reemplaza el último *valor* asociado a la *variable*. [5.5.3]

atributo revelado

E: reveal attribute

F: attribut d'exposition

Una *variable* perteneciente a un *proceso* puede tener un *atributo revelado*, en cuyo caso se permite que otro *proceso* en el *mismo bloque* vea el *valor* asociado con la *variable*. Véase *definición de visión*. [2.6.1.1]

axioma

E: axiom

F: axiome

Un *axioma* es una clase especial de *ecuación* que implica una equivalencia con el *literal booleano Verdadero*. «*Axiomas*» se utiliza como sinónimo de «*axiomas y ecuaciones*». [5.1.3]

bloque

E: block

F: bloc

Un *bloque* es una parte de una *partición* de un *sistema* o de un *bloque* progenitor. Cuando se utiliza solo, *bloque* es sinónimo de *instancia de bloque*. Un *bloque* es una *unidad de ámbito* y proporciona un interfaz estático. [2.4.3]

boolean, véase booleano

booleano ; boolean

E: *boolean*

F: *boléen*

Booleano es un género definido en una *definición parcial de tipo* predefinida y tiene los valores Verdadero y Falso. Para el género *booleano* los operadores predefinidos son NOT, AND, OR, XOR e implicación. [5.6.1]

cadena ; string

E: *String*

F: *chaîne (string)*

Cadena es un generador predefinido utilizado para introducir listas. Los operadores predefinidos incluyen longitud (length), primero (first), último (last), subcadena (substring) y concatenación. [5.6.3]

cadena-de-caracteres ; charstring

E: *Charstring*

F: *chaîne de caractères (character string)*

Cadena-de-caracteres es un género definido en una *definición parcial de tipo* predefinida, para el cual los valores son *cadena de caracteres* y los operadores son los del generador predefinido *cadena* instanciado para caracteres. [5.6.4]

cadena de transición

E: *transition string*

F: *chaîne de transition*

Una *cadena de transición* es una secuencia de cero o más acciones. [2.6.7.1]

calificador

E: *qualifier*

F: *partie qualificative (qualificatif)*

El *calificador* es una parte de un *identificador* que constituye la información adicional a la parte *nombre* del *identificador* para garantizar la unicidad. Los *calificadores* siempre están presentes en la *sintaxis abstracta* pero en la *sintaxis concreta*, sólo han de utilizarse en la medida que lo requiera la unicidad cuando el *calificador* de un *identificador* no pueda derivarse del contexto de utilización de la parte *nombre*. [2.2.2]

canal

E: *channel*

F: *canal*

Un *canal* es la conexión que transporta *señales* entre dos *bloques*. Los *canales* transportan también *señales* entre un *bloque* y el *entorno*. Los *canales* pueden ser unidireccionales o bidireccionales. [2.5.1]

carácter ; character

E: *Character*

F: *caractère (character)*

Carácter es un género definido en una *definición parcial de tipo* predefinida, para el cual los valores son los elementos del alfabeto N.º 5 del CCITT (por ejemplo, 1, A, B, C, etc.). Para el género *carácter* los operadores de *ordenación* están predefinidos. [5.6.2]

character, véase carácter

charstring, véase cadena-de-caracteres

clase de entidad

E: *entity class*

F: *classe d'entité*

Una *clase de entidad* es una categorización de los tipos *LED* basada en la similitud de uso. [2.2.2]

comentario

E: comment

F: commentaire

Un *comentario* es una información completa o aclara la *especificación LED*. En *LED/GR* se puede asociar *comentarios* a cualquier *símbolo* mediante una línea de trazo discontinuo. En *LED/PR* pueden introducirse *comentarios* por medio de la palabra clave *COMMENT*. Los *comentarios* no tienen un significado definido en *LED*. Véase también *nota*. [2.2.6]

comportamiento

E: behavior

F: comportement

El *comportamiento* o *comportamiento funcional* de un *sistema* es el conjunto de secuencias de respuestas a secuencias de estímulos. [1.1.3]

comportamiento funcional

E: functional behavior

F: comportement fonctionnel

Comportamiento funcional es sinónimo de *comportamiento*.

condición habilitante (o habilitadora)

E: enabling condition

F: condition de validation

Una *condición habilitante* es un medio para aceptar condicionalmente una *señal* para *entrada*. [4.12]

conectar

E: connect

F: connect

Conectar indica la conexión de un *canal* a una o más *rutas de señales*.

conector

E: connector

F: connecteur

Un *conector* es un *símbolo LED/GR* que es un *conector de entrada* o un *conector de salida*. Hay una *línea de flujo* implícita desde los *conectores de salida* hasta el *conector de entrada* asociado en el mismo *proceso* o *procedimiento* identificado por el mismo nombre. [2.6.6]

conector de entrada

E: in-connector

F: connecteur d'entrée

Un *conector de entrada* es un *conector*.

conector de salida

E: out-connector

F: connecteur de sortie

Un *conector de salida* es un *conector*.

conjuntista

E: Powerset

F: mode ensembliste

Conjuntista es el *generador predefinido* utilizado para introducir conjuntos matemáticos. Los *operadores* para *conjuntista* son IN, Incl, Del, unión, intersección y los *operadores de ordenación*. [5.6.9]

conjunto completo de señales de entrada válidas

E: complete valid input signal set

F: ensemble complet de signaux d'entrée valides

El *conjunto completo de señales de entrada válidas* de un proceso es la unión del *conjunto de señales de entrada válidas*, las *señales locales*, las *señales de temporizador* y las *señales implícitas* del *proceso*. [2.2.4]

conjunto de señales de conservación

E: save signal set

F: ensemble de signaux de mise en réserve

[2.6.5] El *conjunto de señales de conservación* de un *estado* es el conjunto de *señales* conservadas para ese estado.

conjunto de señales de entrada válidas

E: valid input signal set

F: ensemble de signaux d'entrée valides

El *conjunto de señales de entrada válidas* de un *proceso* es una lista de las *señales* externas manejadas por una *entrada al proceso*. Comprende las *señales de rutas de señales* que conducen al *proceso*. Compárese con *conjunto completo de señales de entrada válidas*. [2.4.4, 2.5.2]

conservación

E: save

F: mise en réserve

Una *conservación* es la declaración de las *señales* que no deben ser consumidas en un *estado* dado. [2.6.5]

crear

E: create

F: créer

Crear es sinónimo de *petición de crear*.

datos predefinidos

E: predefined data

F: données prédéfinies

Para simplificar la descripción, la expresión *datos predefinidos* se aplica tanto a los *nombres* predefinidos para los *géneros* introducidos por *definiciones parciales de tipo* como a los *nombres* predefinidos para *generadores de tipo de datos*. *Booleano*, *carácter*, *cadena de caracteres*, *duración*, *entero*, *natural*, *PID*, *real* y *tiempo* son *nombres de género* predefinidos. *Matriz*, *conjuntista* y *cadena* son *nombres* predefinidos de *generador de tipo de datos*. Los *datos predefinidos* están definidos implícitamente a nivel de *sistema* en todos los *sistemas LED*. [5.6]

decisión

E: decisión

F: décision

Una *decisión* es una *acción* dentro de una *transición* que hace una pregunta cuya respuesta puede obtenerse en ese instante y en consecuencia elige una de las *transiciones* que salen de la *decisión* para continuar la interpretación. [2.7.5]

definición de bloque

E: block definition

F: définition de bloc

Una *definición de bloque* es la definición de un *bloque* en *LED/PR*. [2.4.2]

definición de canal

E: channel definition

F: définition de canal

Una *definición de canal* es la definición de un *canal* en *LED/PR*. [2.5.1]

definición de macro

E: macro definition

F: définition de macro

Una *definición de macro* es la definición de una *macro* en *LED/PR*. [4.2.2]

definición de procedimiento

E: procedure definition

F: définition de procédure

Una *definición de procedimiento* es la definición *LED/PR* de un *procedimiento*. [2.4.5]

definición de proceso

E: process definition

F: définition de processus

Una *definición de proceso* es la representación *LED/PR* de un *proceso*. [2.4.4]

definición de señal

E: signal definition

F: définition de signal

Una *definición de señal* define un *tipo de señal nombrado* y asocia una lista de cero o más *identificadores de género al nombre señal*. Esto permite a las *señales* vehicular *valores*. [2.5.4]

definición de servicio

E: service definition

F: définition de service

Una *definición de servicio* es la definición *LED/PR* de un *servicio*. [4.10.1]

definición de sistema

E: system definition

F: définition de système

Una *definición de sistema* es la representación *LED/PR* de un *sistema*. [2.4.2]

definición de subestructura de bloque

E: block substructure definition

F: définition de sous-structure de bloc

Una *definición de subestructura de bloque* es la representación *LED/PR* de una *subestructura de bloque* para un *bloque particionado*. [3.2.2]

definición de subestructura de canal

E: channel substructure definition

F: définition de sous-structure de canal

Una *definición de subestructura de canal* es la definición de la *subestructura de canal* en *LED/PR*. [3.2.3]

definición de tipo

E: type definition

F: définition de type

Una *definición de tipo* define las propiedades de un *tipo*. [1.3.1]

definición de tipo de datos

E: data type definition

F: définition de type de données

Una *definición de tipo de datos* define la validez de *expresiones* y la relación entre *expresiones* en cualquier punto dado en una *especificación LED*. [5.2.1]

definición de variable

E: variable definition

F: définition de variable

Una *definición de variable* es la indicación de que los *nombres de variable* enumerados serán *visibles* en el *proceso, procedimiento o servicio* que contiene la definición. [2.6.1.1]

definición de visión

E: view definition

F: définition de visibilité

Una *definición de visión* define un *identificador de variable* de otro *proceso* donde tiene el *atributo revelado*. Esto permite al *proceso que observa acceder* al valor de esa *variable*. [2.6.1.2]

definición parcial de tipo

E: partial type definition

F: définition partielle de type

La *definición parcial de tipo* para un *género* define algunas de propiedades relacionadas con el *género*. Una *definición parcial de tipo* es parte de una definición de *tipo de datos*. [5.2.1]

definición remota

E: remote definition

F: définition distante

Una *definición remota* es un medio sintáctico para distribuir una *definición de sistema* entre varias partes y relacionar las partes entre sí. [2.4.1]

descripción

E: description

F: description

Una *descripción* de un *sistema* es la descripción de su *comportamiento* efectivo. [1.1]

diagrama

E: diagram

F: diagramme

Un *diagrama* es la representación *LED/GR* de una parte de una *especificación*. [2.4.2]

diagrama de árbol de bloques

E: block tree diagram

F: diagramme d'arborescence de bloc

Un *diagrama de árbol de bloques* es un documento auxiliar en *LED/GR* que representa la *partición* de un sistema en *bloques a niveles de abstracción* inferiores por medio de un diagrama de árbol invertido (es decir, con el *bloque* progenitor en la parte superior). [3.2.2]

diagrama de bloque

E: block diagram

F: diagramme de bloc

El *diagrama de bloque* es la definición de un *bloque* en *LED/GR*. [2.4.3]

diagrama de flujo de control

E: control flow diagram

F: diagramme de liaison de contrôle

Un *diagrama de flujo de control* es un *diagrama de proceso*, un *diagrama de procedimiento* o un *diagrama de servicio*.

diagrama de interacción

E: interaction diagram

F: diagramme d'interaction

Un *diagrama de interacción* es un *diagrama de bloque*, *diagrama de sistema*, *diagrama de subestructura de canal*, o *diagrama de subestructura de bloque*.

diagrama de macro

E: macro diagram

F: diagramme de macro

Un *diagrama de macro* es la definición de una *macro* en *LED/GR*. [4.2.2]

diagrama de procedimiento

E: procedure diagram

F: diagramme de procédure

Un *diagrama de procedimiento* es la representación *LED/GR* de un *procedimiento*. [2.4.5]

diagrama de proceso

E: process diagram

F: diagramme de processus

Un *diagrama de proceso* es la representación *LED/GR* de la definición de un *proceso*. [2.4.4]

diagrama de servicio

E: service diagram

F: diagramme de service

Un *diagrama de servicio* es la definición *LED/GR* de un *servicio*. [4.10]

diagrama de sintaxis

E: syntax diagram

F: diagramme de syntaxe

Los *diagramas de sintaxis* son ilustraciones de las definiciones de la *sintaxis textual concreta*. [Anexo C2]

diagrama de sistema

E: system diagram

F: diagramme de système

Un *diagrama de sistema* es la representación *LED/GR* de un *sistema*. [2.4.2]

diagrama de subestructura de bloque

E: block substructure diagram

F: diagramme de sous-structure de bloc

Un *diagrama de subestructura de bloque* es la representación *LED/GR* de una *subestructura de bloque* para un *bloque fraccionado*. [3.2.2]

diagrama de subestructura de canal

E: channel substructure diagram

F: diagramme de sous-structure de canal

Un *diagrama de subestructura de canal* es la definición de la *subestructura de canal* en *LED/GR*. [3.2.3]

duración ; duration

E: duration

F: durée (duration)

Duración es un género definido en una *definición parcial de tipo* predefinida, para el cual los *valores* están denotados como *reales* y representan el intervalo entre dos instantes de tiempo. [5.6.11]

duration, véase *duración*

ecuación

E: equation

F: équation

Una *ecuación* es una relación entre *términos* del mismo género que se cumple para todos los *valores* posibles de cada *identificador de valor de la ecuación*. Una *ecuación* puede ser un *axioma*. [5.1.3, 5.2.3]

EMISOR, véase SENDER

entero ; integer

E: integer

F: entier (integer)

Entero es un género definido en una *definición parcial de tipo* predefinida, para el cual los valores son los de los enteros matemáticos (... , -2, -1, 0, +1, +2, ...). Para el género *entero*, los operadores predefinidos son +, -, */ y los operadores de ordenación. [5.6.3]

entorno

E: environment

F: environnement

El término *entorno* es sinónimo de *entorno de un sistema*. Además, cuando el contexto lo permita, puede ser sinónimo de *entorno de un bloque, proceso, procedimiento o servicio*. [1.3.2]

entorno de un sistema

E: environment of a system

F: environnement d'un système

El *entorno de un sistema* es el mundo exterior del *sistema* que se especifica. El *entorno* interactúa con el *sistema* enviando/recibiendo *instancias de señal a/desde el sistema*. [1.3.2]

entrada

E: input

F: entrée

Una *entrada* es el consumo de una *señal aplicada al puerto de entrada* que inicia una *transición*. Durante el consumo de una *señal*, los valores asociados a la *señal* se hacen disponibles a la *instancia de proceso*. [2.6.4, 4.10.2]

error

E: error

F: erreur

Se produce un *error* durante la interpretación de una *especificación válida* de un *sistema* cuando se viola una de las condiciones dinámicas del *LED*. El *comportamiento* del *sistema* después de producirse un *error* no está definido por el *LED*. [1.3.3]

especificación

E: specification

F: spécification

Una *especificación* es una definición de los requisitos de un *sistema*. Una *especificación* consiste en *parámetros generales* requeridos del sistema y la *especificación funcional* de su *comportamiento* requerido. *Especificación* puede utilizarse también como notación taquigráfica de «*especificación y/o descripción*», por ejemplo, en las expresiones *especificación LED* o *especificación de sistema*. [1.1]

especificación válida

E: valid specification

F: spécification valide

Una *especificación válida* es una *especificación* que cumple la *sintaxis concreta* y las *reglas de formación correcta* estáticas. [1.3.3]

estado

E: state

F: état

Un *estado* es una condición en la cual una *instancia de proceso* puede consumir una señal. [2.6.3]

estructura jerárquica

E: hierarchical structure

F: structure hiérarchique

Una *estructura jerárquica* es una estructura de una *especificación de sistema* en la cual una *partición* y un *refinamiento* permiten diferentes visiones del *sistema* a diferentes *niveles de abstracción*. Las *estructuras jerárquicas* permiten la gestión de *especificaciones del sistema* complejas. Véase también *diagrama del árbol de bloques* [3.1]

etiqueta

E: label

F: étiquette

Una *etiqueta* es un *nombre* seguido del carácter «:» y se utiliza para fines de conexión en la *sintaxis textual concreta*. [2.6.6]

exportación

E: export

F: export

El término *exportación* es sinónimo de *operación de exportación*.

exportador

E: exporter

F: exportateur

Un *exportador* de una variable es la *instancia de proceso* que posee la *variable* y *exporta* su *valor*. [4.13]

expresión

E: expression

F: expression

Una *expresión* es un *literal*, una aplicación de *operador*, un *sinónimo*, un *acceso a variable*, una *expresión condicional* o un *operador imperativo* aplicado a una o más *expresiones*. Cuando se interpreta una *expresión* se obtiene un *valor* (o el *sistema* está en *error*). [5.4.2.1]

expresión condicional

E: conditional expression

F: expression conditionnelle

Una *expresión condicional* es una *expresión* que contiene una *expresión booleana* que controla si es interpretada la *expresión consecuencia* o la *expresión alternativa*. [5.5.2.3]

expresión de visión

E: view expression

F: expression de vue

Una *expresión de visión* se utiliza dentro de una *expresión* para dar el *valor* en curso de una *variable vista* («*visionada*»). [5.5.4.4]

expresión fundamental

E: ground expression

F: expression close

Una *expresión fundamental* es una *expresión* que contiene solamente *operadores*, *sinónimos* y *literales*. [5.4.2.2]

expresión simple

E: simple expression

F: expression simple

Una *expresión simple* es una *expresión* que sólo contiene *operadores*, *sinónimos* y *literales* de los *géneros* predefinidos. [4.3.2]

extract!, véase *extraer!*

extraer!; extract!

E: extract!

F: extract!

Extraer! es un *operador* que está implícito en una *expresión* cuando una *variable* va seguida inmediatamente por una o más *expresiones* entre paréntesis. [5.4.2.4, 5.6.8]

FBN (forma Backus-Naur)

E: BNF (Backus-Naur Form)

F: forme BNF (Backus-Naur Form)

La forma Backus-Naur (FBN) es una notación formal utilizada para expresar la *sintaxis textual concreta* de un lenguaje. Se utiliza una forma ampliada de *FBN* para expresar la *gramática gráfica concreta*. [1.5.2, 1.5.3]

generador

E: generator

F: générateur

Un *generador* es una descripción de *neotipo* incompleta. Antes de que un *generador* asuma la condición de *neotipo*, deberá ser instanciado, para lo cual se proporcionará la información que falta. [5.4.1.12]

género

E: sort

F: sorte

Un *género* es un conjunto de *valores* con características comunes. Los *géneros* son siempre no vacíos y disjuntos. [2.3.3, 5.1.3]

género estructurado

E: structured

F: sorte structurée

Un *género estructurado* es un *género* con *operadores* y *ecuaciones* implícitos y una *sintaxis concreta* especial para estos *operadores* implícitos. El *género estructurado* se utiliza para formar *valores* con los denominados campos. Se puede *acceder a los valores* de los campos y *modificarlos* independientemente. [5.4.1.10]

gráfico

E: graph

F: graphe

En la *sintaxis abstracta*, un gráfico es una parte de una *especificación LED*, por ejemplo, un *gráfico de procedimiento* o un *gráfico de proceso*.

gráfico de procedimiento

E: procedure graph

F: graphe de procédure

Un *gráfico de procedimiento* es un no terminal en la *sintaxis abstracta* que representa un *procedimiento*. [2.4.5]

gráfico de proceso

E: process graph

F: graphe de processus

Un *gráfico de proceso* es un no terminal en la *sintaxis abstracta* que representa un *proceso*. [2.4.4]

gramática abstracta

E: abstract grammar

F: grammaire abstraite

La *gramática abstracta* define la *semántica del LED*. La *gramática abstracta* está descrita por la *sintaxis abstracta* y las *reglas de formación correcta*. [1.2, 1.4.1]

gramática concreta

E: concrete grammar

F: grammaire concrète

Una *gramática concreta* es la *sintaxis concreta* junto con las *reglas de formación correcta* para esa *sintaxis concreta*. *LED/GR* y *LED/PR* son las *gramáticas concretas de LED*. Las *gramáticas concretas* se hacen corresponder con la *gramática abstracta* para determinar su *semántica*. [1.2]

gramática gráfica concreta

E: concrete graphical grammar

F: grammaire graphique concrète

La *gramática gráfica concreta* es la *gramática concreta* para la parte gráfica de *LED/GR*. [1.2]

gramática textual común

E: common textual grammar

F: grammaire textuelle commune

La *gramática textual común* es el subconjunto de la *gramática textual concreta* que se aplica tanto al *LED/GR* como al *LED/PR*. [1.2]

hacer!; make!

E: make!

F: make!

Hacer! es una operación que sólo se utiliza en definiciones de *tipos de datos* para formar un *valor* de un tipo complejo (por ejemplo, *género estructurado*). [5.4.1.10, 5.6.8]

identificador

E: identifier

F: identificateur

Un *identificador* es la identificación única de un objeto; consta de un *nombre* y una *parte calificativa*. [2.2.2]

importación

E: import

F: import

El término *importación* es sinónimo de *operación de importación*. [4.13]

importador

E: importer

F: importateur

Un *importador* de una *variable importada* es la *instancia de proceso* que *importa* el *valor*. [4.13]

indefinido

E: undefined

F: indéfini (undefined)

Indefinido es un *valor* «especial» de cada *género*, que indica que a una *variable* de ese *género* todavía no se le ha *asignado* un *valor* normal. Véase *acceder*. [5.5.2.2]

inicializar ; poner

E: set

F: set (initialisation)

Operación definida para *temporizadores*, que permite activarlos. [2.8]

instancia

E: instance

F: instance

Una *instancia* de un *tipo* es un objeto que tiene las propiedades del *tipo* (establecidas en la definición). [1.3.1]

instancia de proceso

E: process instance

F: instance de processus

Una *instancia de proceso* es una *instancia* creada dinámicamente de un *proceso*. Véanse *SELF*, *SENDER*, *PARENT* y *OFFSPRING*. [2.4.4]

instanciación

E: instantiation

F: instantiation

Instanciación es la creación de una *instancia* de un *tipo*. [1.3.1]

integer, véase entero**LED (lenguaje de especificación y descripción del CCITT)**

E: SDL (CCITT Specification and Description Language)

F: LDS (langage de description et de spécification du CCITT)

El *LED (lenguaje de especificación y descripción)* del CCITT es un lenguaje formal que proporciona un conjunto de construcciones para la *especificación* de la funcionalidad de un *sistema*. [1.1]

LED básico

E: basic SDL

F: LDS de base

El *LED básico* es el subconjunto del *LED* definido en el § 2 de la Recomendación Z.100.

LED/EP

E: SDL/PE

F: LDS/PE

El *LED/EP* es un conjunto de elementos pictográficos que pueden utilizarse junto con el símbolo de estado del *LED/GR*. [Anexo E]

LED/GR

E: SDL/GR

F: LDS/GR

El *LED/GR* es la representación gráfica en *LED*. La *gramática* para *LED/GR* está definida por la *gramática gráfica concreta* y la *gramática textual común*. [1.2]

LED/PR

E: SDL/PR

F: LDS/PR

El *LED/PR* es la representación de frase textual en *LED*. La *gramática* del *LED/PR* está definida por la *gramática textual concreta*. [1.2]

línea de flujo

E: flow line

F: ligne de liaison

Una *línea de flujo* es un *símbolo* utilizado para conectar *áreas* en un *diagrama de flujo del control*. [2.2.4, 2.6.7.2.2]

lista de parámetros efectivos

E: actual parameter list

F: liste de paramètres réels

Una *lista de parámetros efectivos* es la lista que enumera los *parámetros efectivos*. Los *parámetros efectivos* están asociados por posición con los elementos respectivos de la *lista de parámetros formales* correspondiente.

lista de parámetros formales

E: formal parameter list

F: liste de paramètres formels

Una *lista de parámetros formales* es una *lista que enumera parámetros formales*.

lista de señales

E: signal list

F: liste de signaux

Una *lista de señales* es una lista de *identificadores de señal* utilizados en *definiciones de canal* y de *ruta de señales* para indicar todas las *señales* que pueden ser transportadas por el *canal* o la *ruta de señales* en un sentido. [2.5.5]

literal

E: literal

F: littéral

Un *literal* denota un *valor*. [2.3.3, 5.1.2, 5.4.1.14]

llamada a (de) macro

E: macro call

F: appel de macro

Una *llamada a macro* es una indicación de un lugar en el cual debe expandirse la *definición de macro* que tiene el mismo *nombre*. [4.2.3]

llamada a (de) procedimiento

E: procedure call

F: appel de procédure

Una *llamada a procedimiento* es la invocación de un *procedimiento* que tiene un *nombre*, con el fin de efectuar la interpretación del *procedimiento* y pasar a éste *parámetros efectivos*. [2.7.3]

macro

E: macro

F: macro

Una *macro* es una colección de elementos sintácticos que tiene un *nombre* y reemplaza a la *llamada a macro* antes de que se considere el significado de la representación *LED* (es decir, una *macro* sólo tiene significado cuando es reemplazada en un contexto determinado). [4.2]

make, véase hacer!

matriz

E: array

F: tableau (array)

Matriz es el *generador* predefinido utilizado para introducir el concepto de matrices, para una mayor comodidad de la definición de matrices.

Meta IV

E: Meta IV

F: Meta IV

Meta IV es una notación formal para expresar la *sintaxis abstracta* de un lenguaje. [1.5.1]

Mismo, véase SELF

modelo

E: model

F: modèle

Un *modelo* da la relación de correspondencia para notaciones *taquigráficas* expresadas en términos de una *sintaxis concreta* previamente definida. [1.4.1, 1.4.2]

modificar!; modify!

E: modify!

F: modifier!

Modificar! es un *operador* que viene implícito en expresiones cuando una *variable* va seguida inmediatamente de expresiones entre paréntesis y, después, de $:=$. Dentro de axiomas, *modificar!* se utiliza explícitamente (véase *extraer!*). [5.4.1.10, 5.6.8]

modify!, véase modificar!

natural

E: Natural

F: naturel

Natural es un *sintipo* definido en una *definición parcial de tipo* predefinida para el cual los *valores* son enteros no negativos (esto es, 0, 1, 2, ...). Los *operadores* son los *operadores* del *género entero*. [5.6.6]

neotipo

E: *newtype*

F: *nouveau type (newtype)*

Un *neotipo* introduce un *género*, un conjunto de *operadores* y un conjunto de *ecuaciones*. Obsérvese que el término *neotipo* podría dar lugar a cierta confusión porque lo que en realidad se introduce es un nuevo *género*; no obstante, se mantiene *neotipo* por razones históricas. [5.2.1]

nivel

E: *level*

F: *niveau*

El término *nivel* es sinónimo de *nivel de abstracción*.

nivel de abstracción

E: *level of abstraction*

F: *niveau d'abstraction*

Un *nivel de abstracción* es uno de los niveles de un *diagrama de árbol de bloques*. Una descripción de un *sistema* es un *bloque* en el *nivel de abstracción* más elevado y se muestra como un solo *bloque* en la parte superior de un *diagrama de árbol de bloques*. [3.2.1]

nodo

E: *node*

F: *nœud*

En la *sintaxis abstracta*, un *nodo* es una designación de uno de los conceptos básicos del *LED*.

nombre

E: *name*

F: *nom*

Un *nombre* es una *unidad léxica* utilizada para nombrar objetos *LED*. [2.2.1, 2.2.2]

nota

E: *note*

F: *note*

Una *nota* es un texto delimitado por */** y **/* que no tiene una semántica definida en *LED*. Véase *comentario*. [2.2.1]

notación taquigráfica (o abreviada)

E: *shorthand notation*

F: *notation abrégée*

Una *notación taquigráfica* es una notación de *sintaxis concreta* que proporciona una representación más compacta que hace referencia implícita a conceptos de *LED básico*. [1.4.2]

null; nulo

E: *Null*

F: *null*

Null es el literal del género *Pid*. [5.6.10]

nulo, véase *null*

OFFSPRING; VÁSTAGO

E: *OFFSPRING*

F: *DESCENDANT (OFFSPRING)*

OFFSPRING es una *expresión* de género *Pid*. Cuando *OFFSPRING* se evalúa en un *proceso*, da el valor *Pid* del *proceso* creado más recientemente por este *proceso*. Si el *proceso* no ha creado ningún *proceso*, el resultado de la evaluación de *OFFSPRING* es *null*. [2.4.4, 5.5.4.3]

opción

E: option

F: option

Una *opción* es una construcción de *sintaxis concreta* en una *especificación de sistema LED* genérica que permite elegir diferentes estructuras de *sistema* antes de que el *sistema* sea interpretado. [4.3.3, 4.3.4]

operación de exportación

E: export operation

F: opération d'exportation

Una *operación de exportación* es la operación por la cual el *exportador* revela el *valor* de una *variable*. Véase *operación de importación*. [4.13]

operación de importación

E: import operation

F: opération d'importation

Una *operación de importación* es la operación que da el *valor* de una *variable exportada*. [4.13]

operador

E: operator

F: opérateur

Un *operador* es una denotación para una operación. Los *operadores* se definen en una *definición parcial de tipo*. Por ejemplo +, -, *, /, son *nombres* de *operadores* definidos para el *género entera*. [5.1.2, 5.1.3]

operador imperativo

E: imperative operator

F: opérateur impératif

Un *operador imperativo* es una *expresión, expresión de visión, expresión activa de temporizador, expresión de importación* o una de las *expresiones PID: SELF, PARENT, OFFSPRING* o *SENDER*. [5.5.4]

operador infijo

E: infix operator

F: opérateur infixe

Un *operador infijo* es uno de los *operadores* diádicos predefinidos del *LED* (=, >, OR, XOR, AND, IN, /=, =, >, <, <=, >=, +, -, //, *, /, MOD, REM) que van entre sus dos argumentos. [5.4.1.1]

operadores de ordenación

E: ordering operators

F: opérateurs de relation d'ordre

Los *operadores de ordenación* son <, <=, >, >=. [5.4.1.8]

página

E: page

F: page

Una *página* es uno de los componentes de una partición física de un *diagrama*. [2.2.5]

palabra clave

E: *keyword*

F: *mot clé*

Una *palabra clave* es una *unidad léxica* en la *sintaxis textual concreta*. [2.2.1]

parada

E: *stop*

F: *arrêt*

Una *parada* es una acción que termina una *instancia de proceso*. Cuando una *parada* es interpretada, todas las *variables* pertenecientes a la *instancia de proceso* son destruidas y todas las *señales retenidas* en el *puerto de entrada* dejan de ser accesibles. [2.6.7.2.3]

parámetro efectivo

E: *actual parameter*

F: *paramètre réel*

Un *parámetro efectivo* es una *expresión* que se da a un *proceso o procedimiento* para el *parámetro formal* correspondiente cuando el *proceso o procedimiento* es *creado* (o *llamado*). Obsérvese que en ciertos casos, en una llamada a procedimiento, un *parámetro efectivo* tiene que ser una *variable* (es decir, un tipo particular de *expresión*; véase *variable IN/OUT*). [2.7.2, 2.7.3, 4.2.2]

parámetro formal

E: *formal parameter*

F: *paramètre formel*

Un *parámetro formal* es un *nombre de variable* al cual se asignan *valores efectivos* o que es reemplazado por *variables efectivas*. [2.4.4, 2.4.5, 4.2, 4.10]

parámetros generales

E: *general parameters*

F: *paramètres généraux*

Los *parámetros generales* en una *especificación* y en una *descripción* de un *sistema* se refieren a factores tales como los límites de temperatura, detalles de construcción, capacidad de las centrales, rendimiento, etc., y no están definidos en *LED*. [1.1]

PARENT; PROGENITOR

E: *PARENT*

F: *PARENT*

PARENT es una *expresión Pid*. Cuando un *proceso* evalúa esta *expresión*, el resultado es el *valor Pid* del *proceso* progenitor. Si el *proceso* fue creado en el momento de inicialización del *sistema*, el resultado es *null*. [2.4.4, 5.5.4.3]

partición

E: *partitioning*

F: *subdivision*

Una *partición* es la división de una unidad en componentes menores que, tomados en su conjunto, tienen el mismo *comportamiento* que la unidad inicial. La *partición* no afecta al interfaz estático de una unidad. [3.1, 3.2]

petición de crear

E: *create request*

F: *demande de création*

Una *petición de crear* es la acción que causa la creación y el comienzo de una nueva *instancia de proceso* que utiliza, como una plantilla, un *tipo de proceso* especificado. Los *parámetros efectivos* en la *petición de crear* reemplazan a los *parámetros formales* en el *proceso*. [2.7.2]

pictograma de estado*E: state picture**F: représentation graphique d'état*

Un *pictograma de estado* es un *símbolo de estado* que contiene elementos pictográficos utilizados para ampliar el *LED/GR* a *LED/EP*. [Anexo E]

PId*E: PId**F: PId*

PId es un *género* definido en una *definición parcial de tipo* para el cual hay un *literal*, *null*. *PId* es una abreviatura de *identificador de instancia de proceso* (en inglés *process instance identifier*), y los *valores* de los *géneros* se utilizan para identificar *instancias de proceso*. [5.5.4.3, 5.6.10]

poner, véase inicializar

por defecto*E: default**F: défaut*

La *asignación por defecto* es una denotación de un *valor* asociado inicialmente a cada *variable* del *género* de la cláusula *por defecto*. La cláusula *por defecto* puede aparecer en *definiciones de tipo de datos*. [5.5.3.3]

procedimiento*E: procedure**F: procédure*

Un *procedimiento* es una encapsulación de una parte del *comportamiento* de un *proceso*. Un *procedimiento* está definido en un lugar pero se puede hacer referencia al mismo varias veces dentro del mismo *proceso*. Véanse *parámetro formal* y *parámetro efectivo*. [2.4.5]

proceso*E: process**F: processus*

Un *proceso* es una máquina de estados finitos ampliadas que comunica. La comunicación puede tener lugar vía *señales* o *variables* compartidas. El *comportamiento* de un *proceso* depende del orden de llegada de las *señales* a su *puerto de entrada*. [2.4.4]

PROGENITOR, véase PARENT

puerto de entrada*E: input port**F: port d'entrée*

Un *puerto de entrada* de un *proceso* es una cola que recibe y retiene *señales* en el orden de llegada hasta que las *señales* son consumidas por una *entrada*. El *puerto de entrada* puede contener cualquier número de *señales retenidas*. [2.4.4]

real*E: Real**F: réel*

Real es un *género* definido en una *definición parcial de tipo predefinida* para el cual los *valores* son los números que pueden representarse por un *entero* dividido por otro. Los *operadores* predefinidos para el *género real* tienen los mismos *nombres* que los *operadores* del *género entero*. [5.6.7]

refinamiento

E: refinement

F: raffinement

Refinamiento es la adición de nuevos detalles a la funcionalidad de un *nivel de abstracción* dado. El *refinamiento* de un *sistema* causa un enriquecimiento de su *comportamiento* o de sus capacidades para manejar más tipos de *señales* e información, incluidas las *señales* hacia o desde el *entorno*. Compárese con *partición*. [3.3]

reglas de formación correcta

E: well-formedness rules

F: règles de bonne formation

Las *reglas de formación correcta* son restricciones impuestas a una *sintaxis concreta* que obligan a cumplir condiciones estáticas no expresadas directamente por las reglas de sintaxis. [1.4.1, 1.4.2]

reglas léxicas

E: lexical rules

F: règles lexicales

Las *reglas léxicas* son reglas que definen la manera de formar *unidades léxicas* a partir de caracteres. [2.2.1, 4.2.1]

reinicializar ; reponer

E: reset

F: reset (réinitialisation)

Operación definida para *temporizadores* que permite desactivarlos. Véase *temporizador activo*. [2.8]

reponer, véase reinicializar

retorno

E: return

F: retour

El *retorno* de un *procedimiento* es la transferencia del control al *procedimiento* o *proceso* llamante. [2.6.7.2.4]

retorno de procedimiento

E: procedure return

F: retour de procédure

Retorno de procedimiento es sinónimo de *retorno*.

ruta de señales

E: signal route

F: acheminement de signaux

Una *ruta de señales* indica el flujo de *señales* entre un *tipo de proceso* y otro *tipo de proceso* en el mismo *bloque* o los *canales* conectados al *bloque*. [2.5.2]

salida

E: output

F: sortie

Una *salida* es una *acción* dentro de una *transición* que genera una *instancia de señal*. [2.7.4, 4.10.2]

selección

E: selection

F: sélection

Acción por la cual se proporcionan los *sinónimos externos* necesarios para hacer una *especificación de sistema* específica partiendo de una *especificación de sistema* genérica. [4.3.3]

SELF; MISMO

E: SELF

F: SELF

SELF es una *expresión Pid*. Cuando un *proceso* evalúa esta *expresión*, el resultado es el *valor Pid* de ese *proceso*. *SELF* nunca produce como resultado el *valor null*. Véanse también *PARENT*, *OFFSPRING*, *Pid*. [2.4.4, 5.5.4.3]

semántica

E: semantics

F: sémantique

La *semántica* da significado a una entidad: las propiedades que tiene, la forma de interpretar su *comportamiento* y las condiciones dinámicas que deben cumplirse para que el *comportamiento* de la entidad satisfaga las reglas del *LED*. [1.4.1, 1.4.2]

SENDER; EMISOR

E: SENDER

F: SENDER (émetteur)

SENDER es una *expresión Pid*. Cuando se evalúa, *SENDER* da el *valor Pid* del *proceso* emisor de la *señal* que activó la *transición* en curso. [2.4.4, 2.6.4, 5.5.4.3]

sentencia de asignación

E: assignment statement

F: instruction d'affectation

Una *sentencia de asignación* es una *sentencia* que asigna un *valor* a una *variable*. [5.5.3]

señal

E: signal

F: signal

Una *señal* es una instancia de un *tipo* de *señal* que comunica información a una *instancia de proceso*. [2.5.4]

señal continua

E: continuous signal

F: signal continu

Una *señal continua* es un medio para definir que, cuando en un estado la condición *booleana* asociada se hace Verdadero, la *transición* que sigue a la *señal continua* es interpretada. [4.11]

señal retenida

E: retained signal

F: signal retenu

Una *señal retenida* es una *señal* en el *puerto de entrada* de un *proceso*, es decir una *señal* que ha sido recibida pero no consumida por el *proceso*. [2.4.4]

servicio

E: service

F: service

Un *servicio* es una manera alternativa de especificar un *proceso*. Cada *servicio* puede definir un *comportamiento* parcial de un *proceso*. [4.10]

signatura de operador

E: operator signature

F: signature d'opérateur

Una *signatura de operador* define el *género* (o los *géneros*) de los *valores* a los cuales puede aplicarse el *operador*, y el *género del valor* resultante. [5.2.2]

símbolo

E: symbol

F: symbole

Un *símbolo* es un terminal en las *sintaxis concretas*. Un *símbolo* puede ser una de un conjunto de formas en la *sintaxis gráfica concreta*.

símbolo de ampliación de texto

E: text extension symbol

F: symbole d'extension de texte

Un *símbolo de ampliación de texto* es un contenedor de texto que pertenece al *símbolo gráfico* al cual está asociado el *símbolo de ampliación de texto*. El texto incluido en el *símbolo de ampliación de texto* sigue al texto del *símbolo* al cual está asociado. [2.2.7]

sinónimo

E: synonym

F: synonyme

Un *sinónimo* es un *nombre* que representa un *valor*. [5.4.1.13]

sinónimo externo

E: external synonym

F: synonyme externe

Un *sinónimo externo* es un *género* predefinido cuyo *valor* no está especificado en la *especificación de sistema*. [4.3.1]

sintaxis abstracta

E: abstract syntax

F: syntaxe abstraite

La *sintaxis abstracta* es el medio para describir la estructura conceptual de una *especificación LED* expresada en las *sintaxis concretas* del *LED*, a saber *LED/GR* y *LED/PR*. [1.2]

sintaxis concreta

E: concrete syntax

F: syntaxe concrète

La *sintaxis concreta* para las diversas representaciones del *LED* son los *símbolos* efectivos utilizados para representar el *LED* y la interrelación entre *símbolos* requerida por las reglas sintácticas del *LED*. Las dos *sintaxis concretas* utilizadas en Z.100 son la *sintaxis gráfica concreta* y la *sintaxis textual concreta*. [1.2]

sintaxis gráfica concreta

E: concrete graphical syntax

F: syntaxe graphique concrète

La *sintaxis gráfica concreta* es la *sintaxis concreta* para la parte gráfica del *LED/GR*. La *sintaxis gráfica concreta* se expresa en Z.100 empleando una forma ampliada de *FBN*. [1.2, 1.5.3]

sintaxis textual concreta

E: concrete textual syntax

F: syntaxe textuelle concrète

La *sintaxis textual concreta* es la *sintaxis concreta* para el *LED/PR* y las partes textuales del *LED/GR*. La *sintaxis textual concreta* se expresa en Z.100 empleando *FBN*. [1.2, 1.5.2]

sintipo

E: syntype

F: syntype

Un *sintipo* determina un conjunto de *valores* que corresponde a un subconjunto de los *valores* del *tipo* progenitor. Los *operadores* del *sintipo* son los mismos que los del *tipo* progenitor. [5.4.1.9]

sistema

E: system

F: système

Un *sistema* es un conjunto de *bloques* conectados entre sí y al *entorno* por *canales*. [2.1]

string, véase cadena

subbloque

E: subblock

F: sous-bloc

Un *subbloque* es un *bloque* contenido en otro *bloque*. Se forman *subbloques* cuando un *bloque* es *fraccionado*. [3.2.1, 3.2.2]

subcanal

E: subchannel

F: sous-canal

Un *subcanal* es un *canal* formado cuando un *bloque* es *fraccionado*. Un *subcanal* conecta un *subbloque* a una frontera del *bloque fraccionado*, o un *bloque* a la frontera de un *canal fraccionado*. [3.2.2, 3.2.3]

subconjunto de partición consistente

E: consistent partitioning subset

F: sous-ensemble de subdivision cohérent

Un *subconjunto de partición consistente* es un conjunto de los *bloques* y *subbloques* en una *especificación de sistema* que proporciona una visión completa del *sistema* con partes conexas a un *nivel de abstracción* correspondiente. Así, cuando un *bloque* o *subbloque* está contenido en un *subconjunto de partición consistente*, sus ancestros y colaterales también lo están. [3.2.1]

subconjunto de refinamiento consistente

E: consistent refinement subset

F: sous-ensemble de raffinement cohérent

Un *subconjunto de refinamiento consistente* es un *subconjunto de partición consistente* que contiene todos los *bloques* y *subbloques* que utilizan las *señales* empleadas por cualquiera o cualesquiera de los *bloques* o *subbloques*. [3.3]

subestructura de bloque

E: block substructure

F: sous-structure de bloc

Una *subestructura de bloque* es la *partición* del *bloque* en *subbloques* y *nuevos canales* a un *nivel de abstracción inferior*. [3.2.2]

subestructura de canal

E: channel substructure

F: sous-structure de canal

Una *subestructura de canal* es una *partición* de un *canal* en un conjunto de *canales* y *bloques* a un *nivel de abstracción inferior*. [3.2.3]

subseñal

E: subsignal

F: sous-signal

Una *subseñal* es un *refinamiento* de una *señal*. Una *subseñal* es una *señal* y puede ser a su vez *refinada*. [3.3]

tarea

E: task

F: tâche

Una *tarea* es una *acción*, dentro de una *transición*, que contiene una *secuencia de sentencias de asignación o texto informal*. La interpretación de una *tarea* depende de la *información del sistema* y puede actuar sobre ella. [2.7.1]

temporizador

E: timer

F: temporisateur

Un *temporizador* es un objeto, perteneciente a una *instancia de proceso*, que puede estar *activo* o *inactivo*. Un *temporizador activo* retorna una *señal de temporización* a la *instancia de proceso* propietaria en un instante especificado. Véase también *inicializar* y *reinicializar*. [2.8, 5.5.4.5]

temporizador activo

E: active timer

F: temporisateur actif

Un *temporizador activo* es un *temporizador* que tiene una *señal de temporizador* en el *puerto de entrada* del *proceso* propietario o está fijado para que produzca una *señal de temporizador* en cierto instante futuro. [2.8, 5.5.4.5]

término

E: term

F: terme

Un *término* es sintácticamente equivalente a una *expresión*. Los *términos* sólo se utilizan en *axiomas* y se distinguen de las *expresiones* por razones de claridad. [5.2.3, 5.3.3]

texto informal

E: informal text

F: texte informel

Texto informal es un texto incluido en una *especificación LED* para el cual la *semántica* no está definida por el *LED*, sino por algún otro modelo. Un *texto informal* se escribe entre apóstrofes. [2.2.3]

tiempo ; time

E: Time

F: temps (time)

Tiempo es un género definido en una *definición parcial de tipo* predefinida para el cual los valores se denotan como los valores de *real*. Los operadores predefinidos que utilizan *tiempo* y *duración* son + y -. [5.5.4.1, 5.6.12]

time, véase tiempo

tipo

E: type

F: type

Un *tipo* es un conjunto de propiedades de entidades. Como ejemplos de clases de *tipos* en *LED* cabe citar *bloques, canales, rutas de señales, señales y sistemas*. [1.3.1]

tipo abstracto de datos

E: abstract data type

F: type abstrait de données

Tipo abstracto de datos es un sinónimo de *tipo de datos*. Todos los *tipos de datos LED* son *tipos abstractos de datos*.

tipo de datos

E: data type

F: type de données

Un *tipo de datos* es la definición de conjuntos de valores (*géneros*), un conjunto de operadores que son aplicados a estos valores y un conjunto de reglas algebraicas (*ecuaciones*) que definen el comportamiento cuando se aplican los operadores a los valores. [2.3.1]

transición

E: transition

F: transition

Una *transición* es una secuencia de acciones que se produce cuando una *instancia de proceso* pasa de un estado a otro. [2.6.7.1]

transición implícita

E: implicit transition

F: transition implicite

En la *sintaxis concreta*, una *transición implícita* es iniciada por una *señal* perteneciente al conjunto completo de señales de entrada válidas y no especificada en una *entrada* o *conservación del estado*. Una *transición implícita* no contiene acción y hace retornar directamente al mismo estado. [4.6]

trayecto de comunicación

E: communication path

F: trajet de communication

Un *trayecto de comunicación* es un medio de transporte que vehicula *instancias de señal* desde una *instancia de proceso* o desde el entorno a otra *instancia de proceso* o al entorno. Un *trayecto de comunicación* comprende uno o más trayectos de *canal*, uno o más trayectos de *ruta de señales* o una combinación de ambas clases de trayectos. [2.7.4]

unidad de ámbito

E: scope unit

F: unité de portée

En la *gramática concreta*, una *unidad de ámbito* define el intervalo de *visibilidad* de los *identificadores*. Ejemplos de *unidades de ámbito* incluyen las definiciones de *sistema, bloque, proceso, procedimiento, definiciones parciales de tipo y definiciones de servicio*. [2.2.2]

unidad léxica

E: lexical unit

F: unités lexicales

Las *unidades léxicas* son los *símbolos* terminales de la *sintaxis textual concreta*. [2.2.1]

valor

E: value

F: valeur

Un *valor* de un *género* es uno de los valores que están asociados con una *variable* de ese *género*, y que puede utilizarse con un *operador* que requiere un *valor* de ese *género*. Un *valor* es el resultado de la interpretación de una *expresión*. [2.3.3, 5.1.3]

variable

E: variable

F: variable

Una *variable* es una entidad que pertenece a una *instancia* de *proceso* o *procedimiento* y que puede ser asociada a un *valor* mediante una *sentencia de asignación*. Cuando se *accede* a una *variable*, ésta da el último *valor* que se le ha asignado. [2.3.2]

variable exportada

E: exported variable

F: variable exportée

Una *variable exportada* es una *variable* que puede utilizarse en una *operación de exportación*. [4.13]

variable importada

E: imported variable

F: variable importée

Una *variable importada* es una *variable* utilizada en una *operación de importación*. [4.13]

variable IN

E: IN variable

F: variable «IN»

Una *variable IN* es un atributo de *parámetro formal* que denota el caso en que un *valor* es pasado a un *procedimiento* vía un *parámetro efectivo*. [2.4.5]

variable IN/OUT

E: IN/OUT variable

F: variable «IN/OUT»

Una *variable IN/OUT* es un atributo de *parámetro formal* que denota el caso en que un *nombre de parámetro formal* se utiliza como un sinónimo de la *variable* (esto es, el *parámetro efectivo* tiene que ser una *variable*). [2.4.5]

VÁSTAGO, véase OFFSPRING

visibilidad

E: visibility

F: visibilité

La *visibilidad* de un *identificador* viene dada por las *unidades de ámbito* en que puede utilizarse. Dos definiciones en la misma *unidad de ámbito* y pertenecientes a la misma *clase de entidad* no pueden tener el mismo *nombre*. [2.2.2]

zona, véase área

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsimil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación