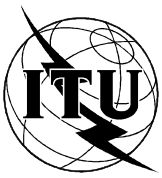

INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

Z.100 Annex B

(11/1988)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

Functional specification and description language (SDL)
Criteria for using formal description techniques (FDTs)

ABSTRACT SYNTAX SUMMARY

Reedition of CCITT Recommendation Z.100 Annex B published in the Blue Book, Fascicle X.1 (1988)

NOTES

1 CCITT Recommendation Z.100 Annex B was published in Fascicle X.1 of the *Blue Book*. This file is an extract from the *Blue Book.* While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

2 In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

ANNEX B
(To Recommendation Z.100)

Abstract syntax summary

Identifier :: *Qualifier Name*

Qualifier = *Path-item* +

Path-item = *System-qualifier* |
Block-qualifier |
Block-Substructure-qualifier |
Signal-qualifier |
Process-qualifier |
Procedure-qualifier |
Sort-qualifier

System-qualifier :: *System-name*

Block-qualifier :: *Block-name*

Block-Substructure-qualifier :: *Block-substructure-name*

Process-qualifier :: *Process-name*

Procedure-qualifier :: *Procedure-name*

Signal-qualifier :: *Signal-name*

Sort-qualifier :: *Sort-name*

Name :: *Token*

Informal-text :: ...

System-definition :: *System-name*
Block-definition-**set**
Channel-definition-**set**
Signal-definition-**set**
Data-type-definition
Syn-type-definition-**set**

System-name = *Name*

Block-definition :: *Block-name*
Process-definition-**set**
Signal-definition-**set**
Channel-to-route-connection-**set**
Signal-route-definition-**set**
Data-type-definition
Syn-type-definition-**set**
[*Block-substructure-definition*]

Block-name = *Name*

Process-definition :: *Process-name*
Number-of-instances
Process-formal-parameter *
Procedure-definition-**set**
Signal-definition-**set**
Data-type-definition
Syn-type-definition-**set**
Variable-definition-**set**
View-definition-**set**
Timer-definition-**set**
Process-graph

Number-of-instances :: *Intg Intg*

Process-name = *Name*

Process-graph :: *Process-start-node*
State-node-**set**

Process-formal-parameter :: *Variable-name*
Sort-reference-identifier

Procedure-definition :: *Procedure-name*
*Procedure-formal-parameter**
Procedure-definition-**set**
Data-type-definition
Syn-type-definition-**set**
Variable-definition-**set**
Procedure-graph

Procedure-name = *Name*

Procedure-formal-parameter = *In-parameter* |
Inout-parameter

In-parameter :: *Variable-name*
Sort-reference-identifier

Inout-parameter :: *Variable-name*
Sort-reference-identifier

Procedure-graph :: *Procedure-start-node*
State-node-**set**

Procedure-start-node :: *Transition*

Channel-definition :: *Channel-name*
Channel-path
[*Channel-path*]

Channel-path :: *Originating-block*

		Destination-block *Signal-identifier*-**set**
Originating-block	=	*Block-identifier* | ENVIRONMENT
Destination-block	=	*Block-identifier* | ENVIRONMENT
Block-identifier	=	*Identifier*
Signal-identifier	=	*Identifier*
Channel-name	=	*Name*
Signal-route-definition	::	*Signal-route-name* *Signal-route-path* [*Signal-route-path*]
Signal-route-path	::	*Originating-process* *Destination-process* *Signal-identifier*-**set**
Originating-process	=	*Process-identifier* | ENVIRONMENT
Destination-process	=	*Process-identifier* | ENVIRONMENT
Signal-route-name	=	*Name*
Channel-to-route-connection	::	*Channel-identifier* *Signal-route-identifier*-**set**
Signal-route-identifier	=	*Identifier*
Signal-definition	::	*Signal-name* *Sort-reference-identifier** [*Signal-refinement*]
Signal-name	=	*Name*
Variable-definition	::	*Variable-name* *Sort-reference-identifier* [REVEALED]
Variable-name	=	*Name*
View-definition	::	*Variable-identifier* *Sort-reference-identifier*
Process-start-node	::	*Transition*

State-node :: *State-name*
Save-signalset
Input-node-**set**

State-name = *Name*

Input-node :: *Signal-identifier*
[*Variable-identifier*]*
Transition

Variable-identifier = *Identifier*

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

Transition :: *Graph-node* *
(*Terminator* | *Decision-node*)

Graph-node :: *Task-node* |
Output -node |
Create-Request-node |
Call-node |
Set-node |
Reset-node

Terminator :: *Nextstate-node* |
Stop-node |
Return-node

Nextstate-node :: *State-name*

Return-node :: ()

Stop-node :: ()

Task-node :: *Assignment-statement* |
Informal -text

Create-request-node :: *Process-identifier*
[*Expression*]*

Process-identifier = *Identifier*

Call-node :: *Procedure-identifier*
[*Expression*] *

Procedure-identifier = *Identifier*

Decision-node :: *Decision-question*
Decision-answer-**set**
[*Else-answer*]

Decision-question = *Expression* |

		Informal-text
Decision-answer	::	*(Range-condition* | *Informal-text) Transition*
Else-answer	::	*Transition*
Output-node	::	*Signal-identifier* [*Expression*]* [*Signal-destination*] *Direct-via*
Signal-destination	=	*Expression*
Direct-via	=	*Signal-route-identifier*-**set**
Timer-definition	::	*Timer-name Sort-reference-identifier**
Timer-name	=	*Name*
Set-node	::	*Time-expression* *Timer-identifier* *Expression**
Reset-node	::	*Timer-identifier* *Expression**
Timer-identifier	=	*Identifier*
Time-expression	=	*Expression*
Block-substructure-definition	::	*Block-substructure-name* *Sub-block-definition*-**set** *Channel-connection*-**set** *Channel-definition*-**set** *Signal-definition*-**set** *Data-type-definition* *Syn-type-definition*-**set**
Block-substructure-name	=	*Name*
Sub-block-definition	=	*Block-definition*
Channel-connection	::	*Channel-identifier* *Sub-channel-identifier*-**set**
Sub-channel-identifier	=	*Channel-identifier*
Channel-identifier	=	*Identifier*
Signal-refinement	::	*Subsignal-definition*-**set**

Subsignal-definition	::	[REVERSE] *Signal-definition*
Data-type-definition	::	*Type-name* *Type-union* *Sorts* *Signature*–**set** *Equations*
Type-union	=	*Type-identifier*–**set**
Type-identifier	=	*Identifier*
Sorts	=	*Sort-name*-**set**
Type-name	=	*Name*
Sort-name	=	*Name*
Equations	=	*Equation*–**set**
Signature	=	*Literal-signature* | *Operator-signature*
Literal-signature	::	*Literal-operator-name* *Result*
Operator-signature	::	*Operator-name* *Argument-list* *Result*
Argument-list	=	*Sort-reference-identifier* +
Result	=	*Sort-reference-identifier*
Sort-reference-identifier	=	*Sort-identifier* | *Syntype-identifier*
Literal-operator-name	=	*Name*
Operator-name	=	*Name*
Sort-identifier	=	*Identifier*
Equation	=	*Unquantified-equation* | *Quantified-equations* | *Conditional-equation* | *Informal-text*
Unquantified-equation	::	*Term* *Term*

Quantified-equations :: *Value-name*–**set**
Sort-identifier
Equations

Value-name = *Name*

Term = *Ground-term* |
Composite-term |
Error-term

Composite-term :: *Value-identifier* |
Operator-identifier Term^{+} |
Conditional-composite-term

Value-identifier = *Identifier*

Operator-identifier = *Identifier*

Ground-term :: *Literal-operator-identifier* |
Operator-identifier Ground-term^{+} |
Conditional-ground-term

Literal-operator-identifier = *Identifier*

Conditional-equation :: *Restriction*–**set**
Restricted-equation

Restriction = *Unquantified-equation*

Restricted-equation = *Unquantified-equation*

Conditional-composite-term = *Conditional-term*

Conditional-ground-term = *Conditional-term*

Conditional-term :: *Condition*
Consequence
Alternative

Condition = *Term*

Consequence = *Term*

Alternative = *Term*

Error-term :: ()

Syntype-identifier = *Identifier*

Syn-type-definition :: *Syntype-name*
Parent-sort-identifier
Range-condition

Syntype-name	=	*Name*
Parent-sort-identfier	=	*Sort-identifier*
Range-condition	::	*Or-operator-identifier* *Condition-item*–**set**
Condition-item	=	*Open-range* | *Closed-range*
Open-range	::	*Operator-identifier* *Ground-expression*
Closed-range	::	*And-operator-identifier* *Open-range* *Open-range*
Or-operator-identifier	=	*Identifier*
And-operator-identifier	=	*Identifier*
Expression	=	*Ground-expression* | *Active-expression*
Ground-expression	::	*Ground-term*
Variable-access	=	*Variable-identifier*
Active-expression	=	*Variable-access* | *Conditional-expression* | *Operator-application* | *Imperative-operator*
Imperative-operator	=	*Now-expression* | *Pid-expression* | *View-expression* | *Timer-active-expression*
Now-expression	::	()
Pid-expression	=	*Self-expression* | *Parent-expression* | *Offspring-expression* | *Sender-expression*
Self-expression	::	()
Parent-expression	::	()
Offspring-expression	::	()
Sender-expression	::	()

View-expression	::	*Variable-identifier* *Expression*
Timer-active-expression	::	*Timer-identifier* *Expression**
Conditional-expression	::	*Boolean-expression* *Consequence-expression* *Alternative-expression*
Boolean-expression	=	*Expression*
Consequence-expression	=	*Expression*
Alternative-expression	=	*Expression*
Operator-application	::	*Operator-identifier* *Expression*$^{+}$
Assignment-statement	::	*Variable-identifier* *Expression*

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	**Languages and general software aspects for telecommunication systems**

Printed in Switzerland

Geneva, 2010