International Telecommunication Union

ITU-T

TELECOMMUNICATION STANDARDIZATION
SECTOR OF ITU

Implementer's guide

(September 2011)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

Formal description techniques (FDT) – Specification and Description Language (SDL)

Specification and Description Language implementer's guide – Version 2.0.0

Source

SDL implementer's guide version 2.0.0 was agreed on 2 September 2011 by ITU-T Study Group 17 (2009-2012)

CONTENTS

Page

1 Introduction 6
1.1 Scope of the Guide 6
1.2 Approval of the Guide 6
1.3 Distribution of the Guide 6
1.4 Contact 6
2 Error reporting procedure 8
2.1 Submission of error reports and change requests 8
2.2 Resolution of errors 8
2.3 Documenting the Resolution of Defects 8
Annex A Change Request Form 9
Annex B Master List of Changes 10
B.1 Objectives and scope 10
B.2 Terminology 10
B.3 Maintenance of Z.100 to Z.109 10
B.4 Z.100 changes 11
B.5 Z.101 changes 11
B.6 Z.102 changes 11
B.7 Z.103 changes 11
B.8 Z.104 changes 11
B.9 Z.105 changes 11
B.10 Z.106 changes 11
B.11 Z.109 changes 11
B.11 Z.111 changes 11
B.11.1 Use of graphical meta-symbols 11
B.12 Z.119 changes 12
B.13 Open issues. 12
B.13.1 Additional issues to be considered 12
B.13.2 List of Open Items 12
B.13.3 List of Closed items (see B.1 for meaning of a "closed" item) 13
Annex C Master list of Changes for SDL-2000 14
C.1 Changes to Z.100 (11/07) 14
C.1.1 Defect correction- 12.3.3.1 Extended variable, Model (see COM 17 -TD 0454 (Geneva, 16-25 September 2009)) 14
C.2 Changes to Z.104 (10/04) 14
C.3 Changes to Z.105 (07/03) 14

C.4 Changes to Z.106 (08/02) 14

C.4.1 Extension - A75 extended task symbol (see COM 17-TD 3202 and COM 17-TD 3225 (Geneva, 10-19 March 2004)) 14

C.5 Changes to Z.109 (06/07) 15

C.5.1 Clarification – 4 Abbreviations and Synonyms - Use of the SDL in references 15

C.5.2 Defect correction – 7.4 DataType, 7.4.2 Constraints – TD0118 item 1 15

C.5.3 Defect correction – 7.13 Signal, 7.13.2 Constraints – TD0118 item 1 15

C.5.4 Defect correction – 8.2 PseudoState - TD0118 item 3.1 15

C.5.5 Defect correction – 8.5 StateMachine, 8.5.2 Constraints - TD0118 item 1 16

C.5.6 Defect correction – 8.6 Transition, 8.6.2 Constraints - TD0118 item 3.1 16

C.5.7 Clarification – 8.6 Transition, 8.6.3 Semantics - TD0118 item 3.1 16

C.5.8 Defect Correction – 9.11 Decision - TD0118 item 3.2 16

C.5.9 Defect Correction – 9.14 If, 9.14.2 Constraints - TD0118 item 3.3 17

C.5.10 Defect Correction – 9.15 LoopNode, 9.14.2 Constraints - TD0118 item 3.3 17

C.5.11 Clarification – 9.16 OpaqueAction - TD0118 item 2 17

C.5.12 Clarification – 9.20 SendSignalAction- TD0118 item 3.2 17

Specification and Description Language implementer’s guide – Version 2.0.0

1 Introduction

This Guide is a compilation of reported defects and their resolutions to the Specification and Description Language ITU-T Recommendations:

- Z.100, Z.101, Z.102, Z.103, Z.104, Z.105, Z.106, Z.109, Z.111 and Z.119.

The Recommendations Z.111 and Z.119 are included in the above list because they are essential normative references. Agreed changes to these documents that have not yet been issued in approved Recommendations are therefore listed here.

This Guide is intended to be an additional authoritative source of information for implementers to be read in conjunction with the Recommendations themselves.

This Guide itself is not an ITU-T Recommendation. However, it records agreed corrections to reported defects.

This Guide is for the SDL-2010 version of the language. The previous Guide version 1.0.2 was for the SDL-2000 version of the language and therefore did not include Z.101, Z.102 and Z.103, and included Z.107 (which was withdrawn in 2008). For convenience, Annex C includes changes to SDL-2000 versions of the relevant superseded Recommendations.

1.1 Scope of the Guide

The Guide records the resolution of defects and maintenance in the following categories as described in Z.100 Appendix II Guidelines for maintenance of the Specification and Description Language:

- errors
- open items
- deficiencies
- clarifications
- modifications
- decommitted features
- extensions

NOTE: This Guide only addresses proposed additions, deletions, or modifications to the Recommendations that are strictly related to maintenance of the Specification and Description Language as described in the Z.100 series. Proposals for new features should be made in the normal way through contributions to ITU-T Study Group 17.

1.2 Approval of the Guide

This Guide is approved by ITU-T Study Group 17.

1.3 Distribution of the Guide

This Guide is available on-line at no charge from the ITU-T (http://www.itu.int/ITU-T/).

1.4 Contact

Any comments should be addressed to the ITU/TSB Secretariat for Study Group 17:

Martin Euchner
ITU/TSB
Place des Nations
CH-1211 Geneva 20
Switzerland

Tel: +41 22 730 5866
Fax: +41 22 730 5853
Email: tsbsg17@itu.int

2 Error reporting procedure

2.1 Submission of error reports and change requests

Any implementer of the Specification and Description Language defined in the Z.100, Z.101, Z.102, Z.103, Z.104, Z.105, Z.106 and Z.109 Recommendations is invited to submit a report using the form found in Appendix II of Z.100 and copied below in Annex A. The report should be submitted to the ITU-T Study Group 17 Secretariat (see clause 1.4). Each form should cover a single error ("error correction") or proposed change. Where the form reports an error, it is important that the form is completed accurately, especially the sections that relate to the base material against which the error report is being raised.

2.2 Resolution of errors

ITU-T Study Group 17 will address the submitted error. Following agreement on a resolution to the error, the proposed resolution will be approved using the appropriate procedures in ITU-T.

Please note that individual responses are not given specifically to those submitting reports, and that the procedure is not intended as a consulting service.

2.3 Documenting the Resolution of Defects

The following ITU-T Recommendations have errors or agreed changes. The defects and their resolutions are recorded in Annex B.

- None

The following ITU-T Recommendations had errors or agreed changes with respect to SDL-2000, but are expected to be replaced by revised Z.100 series Recommendations during the lifetime of this guide. The defects and their resolutions are recorded in Annex C.

- ITU-T Recommendation Z.106

A number of errors or agreed changes with respect to ITU-T Recommendation Z.100 (08/02) were listed in Annex C of version 1.0.2 of this Guide, but were included in ITU-T Recommendation Z.100 (11/07) and therefore are not repeated here.

Annex A

Change Request Form

Change Request Form

Please supply the following details.		
Type of change:	□error correction □ simplification □ modification	□ clarification (or question) □ extension □ decommission
Short summary of change request		
Short justification of the change request		
Is this view shared in your organization?	□ yes	□ no
Have you consulted other users?	□ yes	□ no
How many users do you represent?	□ 1-5 □ 11-100	□6-10 □ over 100
Your name and address		

Please attach further sheets with details if necessary.

SDL (Z.100) Rapporteur, c/o ITU/TSB, Place des Nations, CH-1211 Geneva 20, Switzerland. Fax: +41 22 730 5853, e-mail: tsbmail@itu.int

Annex B

Master List of Changes

This is the master list of changes for the Z.100 (SDL) series Recommendations for approved by the Working Party 3 of Study Group 17 in 2009 according the rules for maintenance in Z.100 itself.

History: The previous version of this list was published in version 1.0.2 of this document, which replaced versions 1.0.1 and 1.0 of this document and the earlier document COM 17-TD 3250 [2001-2004] one of the documents of July 2004 WP C/17 meeting. COM 17-TD 3250 [2001-2004] records the history up to that point, and there seems to be no benefit repeating that historical information in this document.

In accordance with Appendix II to Recommendation Z.100, the information in this document is distributed to users by various means including sdlnews@sdl-forum.org.

B.1 Objectives and scope

The purpose of this document is to record agreed changes to SDL Recommendations (Z.100 to Z.109) and issues that require study and are therefore "open", or have been studied and a decision has been made that the issue is "closed": that is no further study should be undertaken.

The agreed changes come in two categories:

a) Correction of *errors* and *clarifications* (see definitions below and in Z.100, Appendix II);

b) *Extensions* and *modifications* (see definitions below and in Z.100, Appendix II).

The rules for maintenance in an Appendix to Z.100, state that *errors* and *clarifications* published in the Master list of changes "come into effect immediately". Such changes should be published in a Corrigendum, Addendum or revision of the Recommendation as soon as is practical.

Modification and *extensions* imply some change to SDL. The rule in this case is "Unless there are special circumstances requiring such changes to be implemented as soon as possible, such changes will not be recommended until Z.100 is revised."

B.2 Terminology

An *error* is an inconsistency in one or more Recommendations Z.100 to Z.109.

A *textual correction* is a change in the text or diagrams of Recommendations that corrects clerical or typographical errors.

An *open item* is an issue identified but not resolved.

A *deficiency* is an issue identified where the semantics of SDL are not clearly defined in the Recommendations.

A *clarification* is a change to the text (or diagrams) in a Recommendation that does not (intentionally) change the meaning of SDL, but is intended to make the Recommendations less ambiguous or easier to understand.

A *modification* changes the semantics of SDL.

An *extension* is a new feature that does not change the semantics of SDL defined in the approved Recommendations for SDL.

B.3 Maintenance of Z.100 to Z.109

A Z.100 Appendix documents the procedure to be followed for the maintenance of Recommendations Z.100, Z.101, Z.102, Z.103, Z.104, Z.105, Z.106 and Z.109. This procedure requires error corrections, proposed modifications and extensions to be widely publicized and a Master list of changes to be maintained. Clarifications or corrections for errors and deficiencies in the list of changes come into effect "immediately" (that is as soon as the Working Party or Study Group approves the list). Other changes take effect only when the relevant Recommendation is updated.

B.4 Z.100 changes

The document was submitted for approval at the same time as a revised version of the Z.100 series for SDL-2010 was submitted to the AAP procedure for approval. The Z.100 series for SDL-2010 is an update and replaces to the Z.100 series for SDL-2000, and therefore includes corrections and changes to the Z.100 series for SDL-2000. It is therefore expected that shortly after approval of this document there will be an approved Z.100 series and (at the time of issue) no further changes to Z.100. The previous list of changes that applied to Z.100 series for SDL-2000 prior to approval of the Z.100 series for SDL-2010 are listed in Annex C and apply until the corresponding document for SDL-2000 has been superseded.

B.5 Z.101 changes

See B.4 above.

B.6 Z.102 changes

See B.4 above.

B.7 Z.103 changes

See B.4 above.

B.8 Z.104 changes

See B.4 above.

B.9 Z.105 changes

See B.4 above.

B.10 Z.106 changes

See B.4 above. Note that because this document depends on the contents of Z.100 to Z.105, it is possible that this document is approved later than Z.100 to Z.105.

B.11 Z.109 changes

See B.4 above. Note that because this document depends on the contents of Z.100 to Z.105, it is planned that this document is approved later than Z.100 to Z.105.

B.11 Z.111 changes

B.11.1 Use of graphical meta-symbols

The following should be added as an Appendix.

Use of graphical meta-symbols

The meta symbols for the notation of the graphical grammar should always be used according to their definition.

- It is required that the operators (**contains**, **is associated with**, **is followed by**, **is connected to**, **is attached to)** to have a graphical non terminal as the left hand argument, that is a non-terminal ending with the word "symbol".
- The left-hand-side should never be empty.
- The right-hand-side can a set that may be empty, such as the form … **is connected to** {<gate>*}**set** ; that is connected to an unordered (logically and graphically), possibly empty set of <gates>.
- The post-fix operator **set** should be applied consistently. The form {<xxx>*}**set** should be used rather than {{<xxx>*}**set**} or {<xxx>}* **set** as right hand side of a graphical meta operator, where the meaning that the right hand side is an unordered set of <xxx>. In particular {<xxx>}* **set** means an ordered list of <xxx> that is in a set with just one element. In {{<xxx>*}**set**} the outer braces are (probably) redundant. The form {{<xxx>*} <yy> }**set** means an ordered list of <xxx> items with <yy> required in the set but with the placing unordered.
- If **set** is in a graphical context the items are unordered vertically or horizontally but should not overlap.
- If the right hand side of an operator is
  {<xx><yy><zz>} this is an ordered list <xx><yy><zz> and
  {<xx><yy><zz>}**set** should be used if <xx> and <yy> and <zz> are graphical
  {<xx>*} is an ordered list of <xx> and
  {<xx>*}**set** should be used if <xx> is graphical
- For a line symbol that connects two endpoints the form
  <xxx line symbol> **is connected to** {<end1>} **is connected to** {<end2>}
  should be used where possible, though the same meaning is given by
  <xxx line symbol> **is connected to** {<end1> <end2>} **set**.

B.12 Z.119 changes

None.

B.13 Open issues.

B.13.1 Additional issues to be considered

Open issues for which no contributions are available:

- Data type library extensions (predefined object types, Standard Template Library analogue)
- Memory management issues
- Instance sets vs. container types and navigation into composite agents
- Broadcast mechanisms
- Interrupts

B.13.2 List of Open Items

The following is a list of issues classified as open items according to the rules for maintenance for SDL. The list below originates November 1999 but a number of items have been subsequently removed, though at the time of approval of this document the list had not been reviewed for a considerable period and therefore should be revised.

To facilitate the tracking of open items each item is given an identifier of the form (month/year).<number> where month and year identify the meeting at which the item was first put onto the list. For example “(04/97).3”.

To keep the list concise, the details given in relevant documents are not copied to the list, but the documents are referenced. However, a consequence of this procedure is that some references are to temporary documents of meetings, and therefore all referenced documents are kept as ‘retained documents’ on the ITU informal FTP server
(<ftp://username:password@ties.itu.int/u/tsg17/sg17/archive/2001_xchange/wp3/q13/Retained_documents/>)
and on the SDL Forum ftp server at
<ftp:// username:password@ftp.sdl-forum.org/sg17/wp3/q13/Retained_documents/>.

(04/97).13 output -, input *, reset * (TD 34, TD 37 SG 10 April 1997, note that this may also be related to signal as data);

(04/97).17 signal parameter access (TD 34, TD 35 SG 10 April 1997, e.g., refer to signal parameters in enabling conditions);

(04/97).25 more flexible USE syntax – USE p1, p2, p3 (TD 35 SG 10 April 1997);

(10/98).1 Simpler initialization of systems (TD 34 SG 10 April 1997, TDS 603 October 1998,TDN 631 November 1998) and dynamic routing;

(11/99).1 SuperType Method call (R 10 of the study period 1997-2000, SG 10, Annex 10.6, 2.2.13);

(06/00).1 Exit connection points for tasks (TDA03 6.6 June 2000)

B.13.3 List of Closed items (see B.1 for meaning of a “closed” item)

To facilitate the tracking of items each item uses the identifier of the form (month/year).<number> given when the item was first put onto the open item list. For example “(04/97).3”. If the items were never on the open item list, the numbers are consecutive to the open items for the meeting at which the closed item is identified.

(10/96).9 Allow algorithmic operators with external data - requirement not clear (COM-10-1);

(10/96).13 Operators returning sets of values (multivalued operators) - can adequately be handled with STRUCT (COM-10-1);

(04/97).18 signal Priority (TD 34 SG 10 April 1997);

NOTE: In SDL-2010 it is possible to state the availability time for signals.

(04/97).30 Relaxation of the rules for signals to services, because this would break the model for services and context parameters. (TD 35 SG10 April 1997);

NOTE: services are replaced by state aggregation.

(04/97).31 **virtual** as default, because this had been extensively discussed (and rejected) when SDL-92 was formulated and has implications on the use of constraints. (TD 37 SG 10 April 1997);

(10/98).2 remote process creation, because this was added to the language, although the need can be satisfied by remote procedure and the state machine of a block and it was decided an additional construct made the language too complex (TDI 608 Internet meeting Autumn 1998);

Annex C

Master list of Changes for SDL-2000

C.1 Changes to Z.100 (11/07)

C.1.1 Defect correction- 12.3.3.1 Extended variable, Model (see COM 17 -TD 0454 (Geneva, 16-25 September 2009))

<is assigned sign> <expression> was missing.

In the Model section change

<indexed variable> is derived concrete syntax for:

to

The concrete syntax form:

<indexed variable> <is assigned sign> <expression>

is derived concrete syntax for:

This corrects the model, and makes it consistent with <field variable>.

C.2 Changes to Z.104 (10/04)

None.

C.3 Changes to Z.105 (07/03)

None.

C.4 Changes to Z.106 (08/02)

The following are changes to Z.106 (08/02):

C.4.1 Extension - A75 extended task symbol (see COM 17-TD 3202 and COM 17-TD 3225 (Geneva, 10-19 March 2004))

This change is to validate some existing CIF produced using an implementation of CIF prior to agreement of Z.106.

Add to the end of section 7.4.19 A19 CIF descriptor the alternative:

"| <extended task symbol: A75>"

Add to section 7.4.49 just before the example the note:

"NOTE <extended task symbol: A79> can be used as an alternative syntax."

After section 7.4.74 insert as a new section

"7.4.75 A75 extended task symbol:

/* CIF Extendedtask <position and size: B20> ***/**
[<text position: B21>]
<left curly bracket>
<statement list>
<right curly bracket>
<end>

Additional information:

See also <task symbol: A49> that describes an alternate syntax to represent task symbols.

Example:

```
/* CIF Task (800,550) */
TASK myVariable := 0;"
```

C.5 Changes to Z.109 (06/07)

The following are changes to Z.109(06/2007).

C.5.1 Clarification – 4 Abbreviations and Synonyms - Use of the SDL in references

The acronym SDL is used in references before paragraph numbers that refer to Z.100. This is clarified by adding after

> SDL Specification and Description Language

the text

> , and in particular in References clauses to the relevant Z.100 series Recommendation for the reference

C.5.2 Defect correction – 7.4 DataType, 7.4.2 Constraints – TD0118 item 1

Delete the unnecessary constraint

- interfaceRealization shall be empty.

C.5.3 Defect correction – 7.13 Signal, 7.13.2 Constraints – TD0118 item 1

Replace the unnecessary constraint by the text

> No additional constraints

C.5.4 Defect correction – 8.2 PseudoState - TD0118 item 3.1

The "else" of decisions had been omitted.

In 8.2.2 Constraints after

> A Transition shall have a non-empty guard property Constraint (a Boolean Expression)

add the text

> and an empty trigger property

and replace

> Each guard of each Transition that is an outgoing property

by

> Each Boolean guard of each Transition that is an outgoing property

In 8.2.3 Semantics in the paragraph starting

> A <<Pseudostate>> Pseudostate with kind == choice

replace "guard" by "Boolean guard" (twice), and add to the end of the paragraph the sentence

> An outgoing property with an “else” guard property is mapped to an *Else-answer* where the *Transition* is mapped in the same way as for a Boolean guard property.

C.5.5 Defect correction – 8.5 StateMachine, 8.5.2 Constraints - TD0118 item 1

The constraint to ensure a StateMachine does not return a result for a *Composite-state-type*, was invalid. Replace the text

> The returnedResult property shall be empty

by

> No ownedParameter property shall have a direction=return

C.5.6 Defect correction – 8.6 Transition, 8.6.2 Constraints - TD0118 item 3.1

An empty trigger property is valid, so delete the constraint

> The trigger property shall not be empty.

C.5.7 Clarification – 8.6 Transition, 8.6.3 Semantics - TD0118 item 3.1

To make it clear how a Transition is handled with it is the outgoing property of a <<Pseudostate>> Pseudostate with kind == choice, add a new paragraph to the start of 8.6.3 Semantics

> A <<Transition>> Transition that is the outgoing property of a <<Pseudostate>> Pseudostate with kind == choice is mapped as defined for Pseudostate in 8.2.3.

and after

> If a <<Transition>> Transition has a non-empty trigger property and non-empty guard property

add

> and is not the outgoing property of a <<Pseudostate>> Pseudostate with kind==choice

C.5.8 Defect Correction – 9.11 Decision - TD0118 item 3.2

To handle the "else' clause of a decision, in 9.11.2 Constraints add the constraints and note

- For every Clause except the “else” Clause, the predecessorClause set shall be empty, so that there is no requirement that any Clause is evaluated before any other Clause (except the “else” Clause). The predecessorClause set for an “else” Clause shall include every other Clause, so that they all have to be evaluated before the “else” Clause. For every Clause except the “else” Clause, the successorClause set shall contain only the “else” Clause if there is one, otherwise the successorClause set shall be empty, because the order of evaluation is never enforced in SDL-2010. The successorClause set of the “else” Clause shall be empty.

NOTE – The “else” Clause is a Clause that is a successor to all others and whose test part always returns true, so that it is only invoked if all others are false (see UML-SS 12.3.18 ConditionalNode).

- The isAssured property shall be true. Therefore either there is shall be an “else” Clause, or for every possible value of the left hand side of the expressions (the *Decision-question*) there shall be at least one test that succeeds.

Also in the second sentence of 9.11.3 Semantics, after "The Clause" add

> set (excluding the “else” Clause)

and at the end of 9.11.3 Semantics add the sentence

> The “else” Clause (if present) defines the *Else-answer*, otherwise there is no *Else-answer*.

Because an asterisk is not allowed for an else part, in 9.11.5 Notation after the syntax for <algorithm answer part>, add the paragraph

> A <range condition> in an <algorithm answer part> shall not contain an <asterisk>.

and in 9.11.5 Notation after the syntax for <question>, insert the note, paragraph and references section that had been wrongly placed at the end of 9.20.5 (delete them from 9.20.5)

> NOTE – The syntax for <range condition> is given in clause 7.12.4, the notation for Property.
>
> The <question> represents the left-hand side of every test expression (the question). The <range condition> of each <algorithm answer part> determines the operator for the expression and the value for the right-hand side of the expression. If the <range condition> consists of a single <open range>, the operator is the operator corresponding to the <equality sign>, <not equals sign>, <less than sign>, <greater than sign>, <less than or equals sign>, or <greater than or equals sign>. Otherwise, <range condition> is evaluated to a set value that contains the values specified by the <range condition> and the operator is the membership operator for the left-hand side of the test being in this set. The type of the set is the set of all possible values of the type of the left-hand side of the test. The <statement> of the <algorithm answer part> represents the body of the Clause with the test. The test for the <algorithm else part> is the question not being in the set of values covered by any of the right-hand sides (that is the test is true only if all other tests are false). The <statement> of the <algorithm else part> represents the body of the else Clause.
>
> **9.11.5 References**
>
> | | | |
> |---|---|---|
> | SDL: | 11.13.5 | Decision |
> | | 11.14.5 | Decision statement |

C.5.9 Defect Correction – 9.14 If, 9.14.2 Constraints - TD0118 item 3.3

Z.109 does not use OpaqueExpression, so replace "OpaqueExpression" by "<<Expression>> Expression" (twice).

C.5.10 Defect Correction – 9.15 LoopNode, 9.14.2 Constraints - TD0118 item 3.3

Z.109 does not use OpaqueExpression, so replace "OpaqueExpression" by "<<Expression>> Expression" (once).

C.5.11 Clarification – 9.16 OpaqueAction - TD0118 item 2

Change the first NOTE to NOTE 1, and add a NOTE 2

> NOTE 2 – The body elements (if any) of an OpaqueAction are not used to derive the SDL-UML semantics of the OpaqueAction, instead the semantics depend on the actual OpaqueAction subtype and its attributes. However, if one element of the language of the OpaqueAction is [ITU-T Z.100], the corresponding body should be the string for the notation of the OpaqueAction defined in textual syntax.

C.5.12 Clarification – 9.20 SendSignalAction- TD0118 item 3.2

Delete the text wrongly placed after the 9.20.5 References, which is moved to 9.11 Decision (see above).

____