International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.100

(06/2021)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

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

Specification and Description Language – Overview of SDL-2010

Recommendation ITU-T Z.100

Documents
Document Sharing Hub

Full document details: ITU-T Z.100 2021

ITU-T Z-SERIES RECOMMENDATIONS

LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

FORMAL DESCRIPTION TECHNIQUES (FDT)	
Specification and Description Language (SDL)	**Z.100–Z.109**
Application of formal description techniques	Z.110–Z.119
Message Sequence Chart (MSC)	Z.120–Z.129
User Requirements Notation (URN)	Z.150–Z.159
Testing and Test Control Notation (TTCN)	Z.160–Z.179
PROGRAMMING LANGUAGES	
CHILL: The ITU-T high level language	Z.200–Z.209
MAN-MACHINE LANGUAGE	
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.349
Data-oriented human-machine interfaces	Z.350–Z.359
Human-machine interfaces for the management of telecommunications networks	Z.360–Z.379
QUALITY	
Quality of telecommunication software	Z.400–Z.409
Quality aspects of protocol-related Recommendations	Z.450–Z.459
METHODS	
Methods for validation and testing	Z.500–Z.519
MIDDLEWARE	
Processing environment architectures	Z.600–Z.609

For further details, please refer to the list of ITU-T Recommendations.

Documents
Document Sharing Hub

Full document details: ITU-T Z.100 2021

Recommendation ITU-T Z.100

Specification and Description Language – Overview of SDL-2010

Summary

Recommendation ITU-T Z.100 introduces the Specification and Description Language, intended for unambiguous specification and description of telecommunication systems. The scope of the Specification and Description Language is elaborated in clause 1. The ITU-T Z.100 series for SDL-2010 together form a reference manual for the language. The objective of this Recommendation is to provide an introductory overview to the language and the rest of the reference manual contained in the ITU-T Z.100 series for SDL-2010. The language introduced in this document is more fully defined in other Recommendations in the ITU-T Z.100 series for SDL-2010.

Coverage

The Specification and Description Language has concepts for behaviour, data description and (particularly for larger systems) structuring. The basis of behaviour description is extended finite state machines communicating by messages. Data description is based on data types for values and objects. The basis for structuring is hierarchical decomposition and type hierarchies. These foundations of the Specification and Description Language are elaborated in the respective main clauses of Recommendations ITU-T Z.101 to ITU-T Z.105 and ITU-T Z.107. A distinctive feature of the Specification and Description Language is the graphical representation. This Recommendation covers the conventions used to define the Specification and Description Language in the ITU-T Z.100 series, rules for conformance and guidance for maintenance of the language.

Applications

Specification and Description Language is applicable within standard bodies and industry. The main application areas for which the Specification and Description Language has been designed are stated in clause 1.2, but the Specification and Description Language is generally suitable for describing reactive systems. The range of application is from requirement description to implementation.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Z.100	1984-10-19		11.1002/1000/2222
1.1	ITU-T Z.100 Annex A	1984-10-19		11.1002/1000/6664
1.2	ITU-T Z.100 Annex B	1984-10-19		11.1002/1000/6665
1.3	ITU-T Z.100 Annex C1	1984-10-19		11.1002/1000/6666
1.4	ITU-T Z.100 Annex C2	1984-10-19		11.1002/1000/6667
1.5	ITU-T Z.100 Annex D	1984-10-19		11.1002/1000/6668
2.0	ITU-T Z.100	1987-09-30	X	11.1002/1000/10954
2.1	ITU-T Z.100 Annex A	1988-11-25		11.1002/1000/6669
2.2	ITU-T Z.100 Annex B	1988-11-25		11.1002/1000/6670
2.3	ITU-T Z.100 Annex C1	1988-11-25		11.1002/1000/6671
2.4	ITU-T Z.100 Annex C2	1988-11-25		11.1002/1000/6672
2.5	ITU-T Z.100 Annex D	1988-11-25	X	11.1002/1000/3646
2.6	ITU-T Z.100 Annex E	1988-11-25		11.1002/1000/6673
2.7	ITU-T Z.100 Annex F1	1988-11-25	X	11.1002/1000/3647
2.8	ITU-T Z.100 Annex F2	1988-11-25	X	11.1002/1000/3648
2.9	ITU-T Z.100 Annex F3	1988-11-25	X	11.1002/1000/3649
3.0	ITU-T Z.100	1988-11-25		11.1002/1000/3153
3.1	ITU-T Z.100 Annex C	1993-03-12	X	11.1002/1000/3155
3.2	ITU-T Z.100 Annex D	1993-03-12	X	11.1002/1000/3156
3.3	ITU-T Z.100 Annex F1	1993-03-12	X	11.1002/1000/3157
3.4	ITU-T Z.100 Annex F2	1993-03-12	X	11.1002/1000/3158
3.5	ITU-T Z.100 Annex F3	1993-03-12	X	11.1002/1000/3159
3.6	ITU-T Z.100 App. I	1993-03-12	X	11.1002/1000/3160
3.7	ITU-T Z.100 App. II	1993-03-12	X	11.1002/1000/3161
4.0	ITU-T Z.100	1993-03-12	X	11.1002/1000/3154
4.1	ITU-T Z.100 (1993) Add. 1	1996-10-18	10	11.1002/1000/3917
5.0	ITU-T Z.100	1999-11-19	10	11.1002/1000/4764
5.1	ITU-T Z.100 (1999) Cor. 1	2001-10-29	17	11.1002/1000/5567
6.0	ITU-T Z.100	2002-08-06	17	11.1002/1000/6029
6.1	ITU-T Z.100 (2002) Amd. 1	2003-10-29	17	11.1002/1000/7091
6.2	ITU-T Z.100 (2002) Cor. 1	2004-08-29	17	11.1002/1000/356
7.0	ITU-T Z.100	2007-11-13	17	11.1002/1000/9262
8.0	ITU-T Z.100	2011-12-22	17	11.1002/1000/11387
8.1	ITU-T Z.100 Annex F1	2000-11-24	10	11.1002/1000/5239
8.2	ITU-T Z.100 Annex F2	2000-11-24	10	11.1002/1000/5576
8.3	ITU-T Z.100 Annex F3	2000-11-24	10	11.1002/1000/5577
8.4	ITU-T Z.100 Annex F1	2015-01-13	17	11.1002/1000/12354
8.5	ITU-T Z.100 Annex F2	2015-01-13	17	11.1002/1000/12355

Documents
Document Sharing Hub

Full document details: ITU-T Z.100 2021

8.6	ITU-T Z.100 Annex F3	2015-01-13	17	11.1002/1000/12356
9.0	ITU-T Z.100	2016-04-29	17	11.1002/1000/12846
9.1	ITU-T Z.100 Annex F1	2016-10-29	17	11.1002/1000/13040
9.2	ITU-T Z.100 Annex F2	2016-10-29	17	11.1002/1000/13041
9.3	ITU-T Z.100 Annex F3	2016-10-29	17	11.1002/1000/13042
9.4	ITU-T Z.100 Annex F1	2018-11-13	17	11.1002/1000/13732
9.5	ITU-T Z.100 Annex F2	2018-11-13	17	11.1002/1000/13733
9.6	ITU-T Z.100 Annex F3	2018-11-13	17	11.1002/1000/13734
10.0	ITU-T Z.100	2019-10-14	17	11.1002/1000/14048
10.1	ITU-T Z.100 Annex F1	2019-10-14	17	11.1002/1000/14049
10.2	ITU-T Z.100 Annex F2	2019-10-14	17	11.1002/1000/14050
10.3	ITU-T Z.100 Annex F3	2019-10-14	17	11.1002/1000/14051
11.0	ITU-T Z.100	2021-06-13	17	11.1002/1000/14670
11.2	ITU-T Z.100 Annex F2	2021-06-13	17	11.1002/1000/14702
11.3	ITU-T Z.100 Annex F3	2021-06-13	17	11.1002/1000/14703

Keywords

Specification and Description Language, SDL-2010, overview, conventions, grammars, type concept, presentation, tool compliance, system behaviour, system data description, system structuring.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

© ITU 2021

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Documents
Document Sharing Hub

Full document details: ITU-T Z.100 2021

Table of Contents

		Page
1	Scope	1
1.1	Objective	1
1.2	Application	1
1.3	System specification	2
2	References	2
3	Definitions	3
4	Abbreviations and acronyms	4
5	Conventions	5
5.1	Specification and Description Language grammars	5
5.2	Basic definitions	5
5.3	Presentation style	7
5.4	Choice of grammar rules and rule names	8
6	Tool compliance	8
6.1	Definitions of valid tools	8
6.2	Conformance	8
7	Allocation of features of SDL-2010 to Recommendations	8
7.1	Basic SDL-2010 – [ITU-T Z.101]	9
7.2	Comprehensive SDL-2010 – [ITU-T Z.102]	9
7.3	Shorthand notation and annotation in SDL-2010 – [ITU-T Z.103]	9
7.4	Data and action language in SDL-2010 – [ITU-T Z.104]	9
7.5	SDL-2010 combined with ASN.1 modules – [ITU-T Z.105]	9
7.6	Common Interchange Format for SDL-2010 – [ITU-T Z.106]	9
7.7	Object-oriented data in SDL-2010 – [ITU-T Z.107]	10
Annex A – Abstract syntax index		11
Annex B – BNF syntax index		17
Annex C – Compatibility		35
Annex D – Data defined in the package Predefined		36
D.1	Rules for "=" (equal), "/=" (not equal), comparison, data signatures and literals	36
D.2	Package Predefined overview	36
Annex E – Reserved for examples		45
Annex F – Formal definition		46
Appendix I – Status of ITU-T Z.100, related documents and Recommendations		47
Appendix II – Guidelines for the maintenance of SDL-2010		48
II.1	Maintenance of SDL-2010	48

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Page

Appendix III – Evolution of the Specification and Description Language..... 51
III.1 Versions of the Specification and Description Language 51
III.2 Differences between SDL-88 and SDL-92..... 51
III.3 Differences between SDL-92 and SDL-2000..... 52
III.4 Differences between SDL-2000 and SDL-2010..... 54
Bibliography..... 57

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Introduction

Status/Stability

This Recommendation is an introduction to the ITU-T Z.100 series of Recommendations for SDL-2010 that give the complete language reference manual for SDL-2010. The main text of this Recommendation is stable. Appendix I records the status of the Recommendation series, and should be updated as further studies are completed. The current language definition is based on wide user experience, recent additional user needs, clarifications and corrections. SDL-2010 as defined in this series of Recommendations should meet most user needs, and is based on a previous version called SDL-2000.

SDL-2000 contained a reference data type (**object type**) feature, but this had a number of complexities including dynamic binding, and tool support was lacking. It was therefore decided that this feature should be removed and further study took place leading to a further Recommendation in 2012 plus updates to other Recommendations in the ITU-T Z.100 series for a revised object-oriented data in SDL-2010 using a reference data type feature.

The main text is accompanied by appendices and annexes:

– Appendix I Status of ITU-T Z.100, related documents and Recommendations;
– Appendix II Guidelines for the maintenance of SDL-2010;
– Appendix III Evolution of the Specification and Description Language;
– Annex A Abstract syntax index;
– Annex B BNF syntax index;
– Annex C Compatibility;
– Annex D Data defined in the package Predefined;
– Annex E Reserved for examples.

The following Annex is published separately:

– Annex F Formal definition.

Annex F is the formal definition for the language and provides a more formal definition for SDL-2010 that provides more detail on issues not covered by the rest of the ITU-T Z.100 series for SDL-2010. If there is an inconsistency between Annex F and ITU-T Z.100 or other parts of the ITU-T Z.100 series for SDL-2010, there is an error in the ITU-T Z.100 series Recommendations and further study is needed to determine the correction.

The ITU-T Z.100 series has also an independently published supplement:

– ITU-T Z.Sup1: ITU-T Z.100 series – Supplement on SDL+ methodology: Use of ITU System Design Languages.

ITU-T Z.Sup1 is based on a methodology for an earlier version of the language. The methodology is still applicable because the earlier language features that are used all exist in SDL-2010. The current language has some additional features that are not fully exploited by ITU-T Z.Sup1.

Associated work

One method for usage within standards is described in Recommendation ITU-T Q.65. A recommended strategy for introducing a formal description technique like the Specification and Description Language in standards is available in Recommendation ITU-T Z.110. The use of the Specification and Description Language is also recommended in Recommendation ITU-T Z.450, Quality aspects of protocol-related Recommendations. For references to additional material on the Specification and Description Language, and information on industrial usage, see http://www.sdl-forum.org.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Background

Different versions of the Specification and Description Language have been recommended by ITU-T since 1976. The SDL-2010 version is a revision of SDL-2000, the last edition of which was published in 2007. SDL-2000 was initially published in Recommendation ITU-T Z.100 (1999) as a revision of Recommendation ITU-T Z.100 (1993) incorporating Addendum 1 to Recommendation ITU-T Z.100 (1996) and parts of Recommendation ITU-T Z.105 (1995). Recommendation ITU-T Z.100 (2002) was a technical update of Recommendation ITU-T Z.100 (1999) that incorporated a number of technical corrections and amendments, and without the textual phrase alternative syntax, which had been moved to Recommendation ITU-T Z.106 (2002).

Compared to the Specification and Description Language as defined in 1992, the versions defined in SDL-2000 and SDL-2010 are extended in the areas of object-oriented data, harmonization of a number of features to make the language simpler and features to enhance the usability of the Specification and Description Language with other languages such as ASN.1 and UML. Other minor modifications have been included. Though care has been taken not to invalidate existing documents using the Specification and Description Language as defined in 1992, it is possible some changes require some descriptions to be updated to use this version. Details on the evolution of the language are in Appendix III.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Recommendation ITU-T Z.100

Specification and Description Language – Overview of SDL-2010

1 Scope

The purpose of recommending the Specification and Description Language is to provide a language for unambiguous specification and description of the behaviour of telecommunication systems. The specifications and descriptions using the language are intended to be formal in the sense that it is possible to analyse and interpret them unambiguously.

The terms specification and description are used with the following meaning:

a) a specification of a system is the description of its required behaviour; and

b) a description of a system is the description of its actual behaviour (that is, its implementation).

A system specification, in a broad sense, is the specification of both the behaviour and a set of general parameters of the system. However, the Specification and Description Language is intended to specify the behavioural aspects of a system; the general parameters describing properties like capacity and weight have to be described using different techniques.

This Recommendation gives an overview of the series of Recommendations that define SDL-2010, defines terms, conventions including meta-languages, tool compliance and the basis of data for SDL-2010.

NOTE – Since there is no distinction between use for specification and its use for description, the term specification is used in the SDL-2010 Recommendations for both required behaviour and actual behaviour.

1.1 Objective

The general objectives when defining the Specification and Description Language have been to provide a language that:

a) is easy to learn, use and interpret;

b) provides unambiguous specification for ordering, tendering and design, while also allowing some issues to be left open;

c) is able to be extended to cover new developments;

d) is able to support several methodologies of system specification and design.

1.2 Application

The Recommendations for SDL-2010 provide the reference manual for the Specification and Description Language. This Recommendation provides an overview of the language and the conventions used to define the language. A methodology framework document, which gives examples of Specification and Description Language usage, is available as the Supplement to the Recommendation ITU-T Z.100 series originally produced in the study period 1992-1996.

The main area of application for the language is the specification of the behaviour of aspects of real-time systems, and the design of such systems. Applications in the field of telecommunications include:

a) call and connection processing (for example, call handling, telephony signalling, metering) in switching systems;

b) maintenance and fault treatment (for example, alarms, automatic fault clearance, routine tests) in general telecommunication systems;

c) system control (for example, overload control, modification and extension procedures);

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

d) operation and maintenance functions, network management;

e) data communication protocols;

f) telecommunication services.

The Specification and Description Language is, of course, usable for the functional specification of the behaviour of any object whose behaviour is specifiable using a discrete model; that is, where the object communicates with its environment by discrete messages.

The Specification and Description Language is a rich language and is usable for both high level informal (and/or formally incomplete) specifications, semi-formal and detailed specifications. The user chooses the appropriate parts of the Specification and Description Language for the intended level of communication and the environment in which the language is being used. Depending on the environment in which a specification is used, it is possible many aspects are left to the common understanding between the provider and the user of the specification.

Thus, the language is used for producing:

a) facility requirements;

b) system specifications;

c) ITU-T Recommendations, or other similar standards (international, regional or national);

d) system design specifications;

e) detailed specifications;

f) system design descriptions (both high level and detailed enough to directly produce implementations);

g) system testing descriptions (in particular in combination with Message Sequence Chart [MSC] and Testing and Test Control Notation [TTCN]).

The user organization is able to choose the appropriate level of application of SDL-2010.

1.3 System specification

A specification using the Specification and Description Language defines system behaviour in a stimulus/response fashion, assuming that both stimuli and responses are discrete and carry information. In particular, a system specification is seen as the sequence of responses to any given sequence of stimuli.

The system specification model is based on the concept of communicating extended finite state machines.

The Specification and Description Language also provides structuring concepts that facilitate the specification of large and/or complex systems. These constructs allow the partitioning of the system specification into manageable units that are capable of being handled and understood independently. It is possible to perform partitioning in a number of steps resulting in a hierarchical structure of units defining the system at different levels.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

[ITU-T T.50] Recommendation ITU-T T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.

[ITU-T Z.101] Recommendation ITU-T Z.101 (2021), *Specification and Description Language – Basic SDL-2010*.

[ITU-T Z.102] Recommendation ITU-T Z.102 (2021), *Specification and Description Language – Comprehensive SDL-2010*.

[ITU-T Z.103] Recommendation ITU-T Z.103 (2021), *Specification and Description Language – Shorthand notation and annotation in SDL-2010*.

[ITU-T Z.104] Recommendation ITU-T Z.104 (2021), *Specification and Description Language – Data and action language in SDL-2010*.

[ITU-T Z.105] Recommendation ITU-T Z.105 (2021), *Specification and Description Language – SDL-2010 combined with ASN.1 modules*.

[ITU-T Z.106] Recommendation ITU-T Z.106 (2021), *Specification and Description Language – Common interchange format for SDL-2010*.

[ITU-T Z.107] Recommendation ITU-T Z.107 (2021), *Specification and Description Language – Object-oriented data in SDL-2010*.

[ITU-T Z.111] Recommendation ITU-T Z.111 (2016), *Notations and guidelines for the definition of ITU-T languages*.

[ISO/IEC 10646] ISO/IEC 10646:2020, *Information technology – Universal Coded Character Set (UCS)*.

3 Definitions

There are numerous terms defined throughout this Recommendation and the rest of the ITU-T Z.100 series for SDL-2010 and a complete list of definitions in this clause or in each of these Recommendations would be a repetition of much of the text of the Recommendations. Therefore, only a few key terms are given in this clause.

This Recommendation defines the following terms:

3.1 agent: The term agent is used to denote a system, block or process that contains one or more extended finite state machines.

3.2 block: A block is an agent that contains one or more concurrent blocks or processes and is also permitted to contain an extended finite state machine that owns and handles data within the block.

3.3 body: A body is a state machine graph of an agent, procedure, composite state, or operation.

3.4 channel: A channel is a communication path between agents.

3.5 environment: The environment of the system is everything in the surroundings that communicates with the system in a Specification and Description Language-like way.

3.6 gate: A gate represents a connection point for communication with an agent type, and when the type is instantiated it determines the connection of the agent instance with other instances.

3.7 instance: An instance is an object created when a type is instantiated.

3.8 ITU-T Z.100 series for SDL-2010: This Recommendation and the associated Recommendations [ITU-T Z.101], [ITU-T Z.102], [ITU-T Z.103], [ITU-T Z.104], [ITU-T Z.105], [ITU-T Z.106], [ITU-T Z.107] and any further Recommendation subsequently added to this series.

3.9 pid: The term pid is used for the sort of data items that identify agent instances.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

3.10 **procedure**: A procedure is an encapsulation of part of the behaviour of an agent, which is defined in one place but is able to be called from several places within the agent. Other agents are able to call a remote procedure.

3.11 **process**: A process is an agent that contains an extended finite state machine, and possibly contains other processes.

3.12 **signal**: The primary means of communication is by signals that are output by the sending agent and input by the receiving agent.

3.13 **sort**: A sort is a set of data items that have common properties.

3.14 **state**: An extended finite state machine of an agent is in a state if it is waiting for a stimulus.

3.15 **stimulus**: A stimulus is an event that is able to cause an agent that is in a state to enter a transition.

3.16 **system**: A system is the outermost agent that communicates with the environment.

3.17 **timer**: A timer is an item owned by an agent that causes a timer signal stimulus to occur at a specified time.

3.18 **transition**: A transition is a sequence of actions an agent performs until it enters a transition terminator such as the next state, a return from a composite state, a return from a procedure, or a decision on the subsequent transition.

3.19 **type**: A type is a definition that is used for the creation of instances, or is inherited and specialized to form other types. A parameterized type is a type that has parameters. When these parameters are given different actual parameters, different unparameterized types are defined that, when instantiated, give instances with different properties.

3.20 **value**: The term value is used for the class of data that is accessed directly. Values are allowed to be freely passed between agents.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BNF	Backus-Naur Form
CIF	Common Interchange Format
IRV	International Reference Version of the International Reference Alphabet as defined in [ITU-T T.50]
MSC	Message Sequence Chart
SDL-88	Specification and Description Language as defined by Recommendation ITU-T Z.100 (1988)
SDL-92	Specification and Description Language as defined by Recommendation ITU-T Z.100 (1993) with Addendum 1 (1996)
SDL-2000	Specification and Description Language as defined by Recommendation ITU-T Z.100 (2007)
SDL-2010	Specification and Description Language as defined by the ITU-T Z.100 series for SDL-2010
TTCN	Testing and Test Control Notation
UCS	Universal Character Set of [ISO/IEC 10646]
UML	Unified Modelling Language

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

5 Conventions

The text of this clause defines the conventions used for describing the Specification and Description Language. The meta-languages and conventions introduced are solely introduced for the purpose of describing Specification and Description Language unambiguously.

The conventions of [ITU-T Z.111] apply to all the Recommendations in the ITU-T Z.100 series for SDL-2010.

5.1 Specification and Description Language grammars

In the ITU-T Z.100 series for SDL-2010 the *Abstract grammar* and *Concrete grammar* (see clause 5.3.2 below) define the Specification and Description Language. The syntax of the *Concrete grammar* is in some cases supplemented by a *Model* (see clause 5.3.2 below). A system specification only conforms to the language if it conforms to these grammars (see clause 5.1 of [ITU-T Z.111]). The way a system specification behaves is defined by *Semantics* (see clause 5.3.2 below).

A formal definition is provided which defines how to transform a system specification into the abstract syntax and defines how to interpret a specification, given in terms of the abstract grammar. The formal definition given in Annex F (published separately) is a comprehensive formal definition for SDL-2010.

5.2 Basic definitions

Some general concepts and conventions are used throughout the ITU-T Z.100 series for SDL-2010; their definitions are given in the following subclauses.

5.2.1 Definition, type and instance

In the ITU-T Z.100 series for SDL-2010, the concepts of type and instance and their relationship are fundamental. The schema and terminology defined below and shown in Figure 5-1 are used.

This subclause introduces the basic semantics of type definitions, instance definitions, parameterized type definitions, parameterization, binding of context parameters, specialization and instantiation.

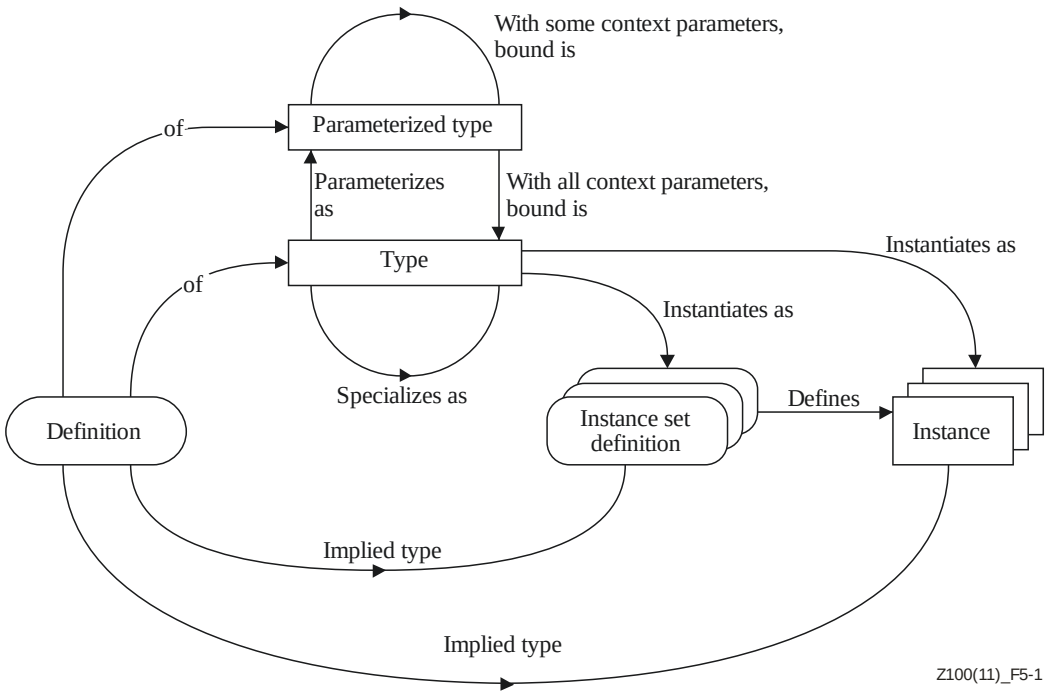

Figure 5-1 – The type concept

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Definitions introduce named entities, which are types or instances with implied types or an instance set that defines the behaviour instances. A definition of a type defines all properties associated with that type. An example of an instance definition is a state definition. An example of a definition that is a type definition is a signal definition. An example of an instance set definition is a process definition. Block and process definitions introduce instance set definitions.

A type is allowed to be instantiated by any number of instances. An instance of a particular type has all the properties defined for that type. An example of a type is a procedure, which is instantiated by procedure calls.

A parameterized type is a type where some entities are represented as formal context parameters. A formal context parameter of a type definition has a constraint. The constraints allow static analysis of the parameterized type. Binding all the parameters of a parameterized type yields an ordinary type. An example of a parameterized type is a parameterized signal definition where one of the sorts conveyed by the signal is specified by a formal sort context parameter; this allows the parameter to be of different sorts in different contexts.

An instance is defined either directly or by the instantiation of a type. An example of an instance is a system instance, which is either defined by a system definition, or is an instantiation of a system type. However, where an instance is defined directly (for example, a system definition), this is actually an instantiation of the (anonymous) type for the direct definition.

Specialization allows one type, the subtype, to be based on another type, its supertype, by adding properties to those of the supertype or by redefining virtual properties of the supertype. A virtual property is allowed to be constrained in order to provide for analysis of general types.

Binding all context parameters of a parameterized type yields an unparameterized type. There is no supertype/subtype relationship between a parameterized type and the type derived from it.

NOTE – To avoid cumbersome text in the ITU-T Z.100 series for SDL-2010, the convention is used that the term "instance" is often omitted. For example, "a system is interpreted..." means "a system instance is interpreted...".

5.2.2 Environment

Systems that are specified in the Specification and Description Language behave according to the stimuli exchanged with the external world. This external world is called the environment of the system being specified.

It is assumed that there are one or more agent instances in the environment, and therefore stimuli flowing from the environment towards the system have associated identities of these agent instances. These agents have pids that are distinguishable from any other pid within the system (see clause D.2.16).

Although the behaviour of the environment is non-deterministic, it is assumed to obey the constraints given by the system specification.

5.2.3 Validity and errors

A system specification is a valid Specification and Description Language system specification only if it satisfies the syntactic rules and the static conditions defined in the ITU-T Z.100 series for SDL-2010.

If a valid Specification and Description Language specification is interpreted and a dynamic condition is violated, then an error occurs. Predefined exceptions (see clause D.2.20) will be raised when an error is encountered during the interpretation of a system. SDL-2010 does not define the handling of exceptions; therefore if an exception occurs the subsequent behaviour of the system cannot be derived from the specification.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

For most cases where an exception might be raised (for example, a range check or incorrect indexing), it is possible to include actions to check before the error is encountered if the exception will be raised, and take appropriate action to avoid the error occurring. Static analysis or dynamic interpretation of a specification might also indicate that it is inevitable an exception is raised, leading to modification of the specification to avoid the situation.

5.3 Presentation style

The conventions of [ITU-T Z.111] apply.

5.3.1 Division of text

The conventions of [ITU-T Z.111] apply.

5.3.2 Titled enumeration items

Abstract grammar

The abstract grammar is specified in the form defined in clause 5.4.1 of [ITU-T Z.111]. The textual presentation of abstract syntax in clause 5.4.1.1 of [ITU-T Z.111] is used.

Concrete grammar

The concrete grammar is specified in the form defined in clause 5.4.2 of [ITU-T Z.111].

The metasymbol ***is followed by*** is used when the left-hand argument is followed by the right-hand argument, and this is shown in diagrams by a <flow line symbol> (see further description in clause 6.5 of [ITU-T Z.101]).

The right-hand argument of the metasymbol ***is associated with*** shall be closer to the left-hand argument than to any other graphical symbol. The syntactical elements of the right-hand argument shall be distinguishable from each other.

Except graphical symbols that are line symbols (such as <flow line symbol>, see the *Concrete grammar* description for the symbol), graphical symbol boundaries shall not overlay or cross. An exception to this rule applies for line symbols, which are allowed to cross each other. There is no logical association between line symbols that cross.

A line symbol consists of a line (solid or dashed) possibly with some additional decorations (typically an arrowhead) at one end or both ends of the line or on the line. The line of a line symbol consists of one or more joined straight-line segments. These segments should normally be horizontal or vertical.

Semantics

All instances have an identity property, but unless this is formed in some unusual way, this identity property is determined as defined by clause 6.6 of [ITU-T Z.101]. This is usually not mentioned as an identity property. Also, it has not been necessary to mention sub-components of definitions contained by the definition since the ownership of such sub-components is obvious from the abstract syntax. For example, it is obvious that a block type definition "has" enclosed processes and/or blocks.

Model

Some constructs are considered to be "derived concrete syntax" (or a shorthand notation) for other equivalent concrete syntax constructs. For example, omitting an input for a signal is derived concrete syntax for an input for that signal followed by a null transition back to the same state (see [ITU-T Z.103]).

Precise details of the order of transformation are found in Annex F.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

5.4 Choice of grammar rules and rule names

The grammar given in the ITU-T Z.100 series for SDL-2010 has been written to aid the presentation in this Recommendation so that the rule names are meaningful in the context they are given and are readable in text. This means that there are a number of apparent ambiguities that are easily resolved by systematic rewriting of the syntax rules or the application of semantic rules.

6 Tool compliance

This clause defines the compliance for tools that claim to support the Specification and Description Language.

The validity of a specification is defined as in clause 5.2.3.

6.1 Definitions of valid tools

6.1.1 compliant SDL-2010 tool: A tool that detects non-compliance of a description with the ITU-T Z.100 series for SDL-2010. If the tool handles a superset notation, it is allowed to categorize non-compliance as a warning rather than a failure.

6.1.2 fully compliant SDL-2010 tool: A compliant SDL-2010 tool that supports the complete grammar defined by the ITU-T Z.100 series for SDL-2010.

6.1.3 valid basic SDL-2010 tool: A compliant SDL-2010 tool that supports the graphical grammar defined in this Recommendation in combination with [ITU-T Z.101].

6.1.4 valid SDL-2010 tool: A compliant SDL-2010 tool that supports the graphical grammar defined in the ITU-T Z.100 series for SDL-2010.

6.1.5 valid SDL-2010 with ASN.1 tool: A valid SDL-2010 tool that also supports ASN.1 as modules according to [ITU-T Z.105].

6.1.6 valid CIF SDL-2010 tool: A compliant SDL-2010 tool that supports the textual SDL-2010 grammar as defined in Level 0 CIF (see clause 5 of [ITU-T Z.106]), which (by definition) includes the semantics and some concrete syntax of other Recommendations in the ITU-T Z.100 series for SDL-2010.

6.1.7 valid CIF SDL-2010 with ASN.1 tool: A valid CIF SDL-2010 tool that also supports ASN.1 as modules according to [ITU-T Z.105].

6.2 Conformance

A conformance statement clearly identifying the language features and requirements not supported should accompany any tool that handles a subset of the language defined by the ITU-T Z.100 series for SDL-2010. If no conformance statement is provided, it shall be assumed that the tool is a fully compliant SDL-2010 tool. It is therefore preferable to supply a conformance statement; otherwise, any unsupported feature allows the tool to be rejected as not valid.

7 Allocation of features of SDL-2010 to Recommendations

The essential behaviour of a system defined using SDL-2010 depends on the extended finite state machine model of [ITU-T Z.101] coupled with the behaviour of expressions of [ITU-T Z.104]. The other Recommendations [ITU-T Z.102], [ITU-T Z.103], [ITU-T Z.105], [ITU-T Z.107] and [ITU-T Z.106] provide language features that (respectively): make the language more comprehensive, make the language easier and more practical to use, provide the full data model and action language (except object-oriented data), enable ASN.1 to be used, define object-oriented data, and define the interchange format. The following outlines the content of [ITU-T Z.101] to [ITU-T Z.107]. The content of Recommendations subsequently added to the ITU-T Z.100 series for SDL-2010 will be described in each Recommendation.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

7.1 Basic SDL-2010 – [ITU-T Z.101]

[ITU-T Z.101] is the ITU-T Recommendation containing the part of the specification of SDL-2010 that covers core features such as agent (block, process) type diagrams containing agent instance structures with channels, diagrams for extended finite state machines and the associated semantics for these basic features. The character set used is International Reference Version (IRV) (see [ITU-T T.50]).

7.2 Comprehensive SDL-2010 – [ITU-T Z.102]

[ITU-T Z.102] is the ITU-T Recommendation containing the part of the specification of SDL-2010 that extends the semantics and syntax of the Basic SDL-2010 language in [ITU-T Z.101] to cover the full abstract grammar and the corresponding canonical concrete notation. This includes features such as continuous signals, enabling conditions, type inheritance and composite states. Also included are features for generic systems, macros and the handling of Universal Character Set (UCS) (see [ISO/IEC 10646]), though these do not need additional abstract grammar.

7.3 Shorthand notation and annotation in SDL-2010 – [ITU-T Z.103]

[ITU-T Z.103] is the ITU-T Recommendation containing the part of the specification of SDL-2010 that adds shorthand notations (such as asterisk state) that make the language easier to use and more concise, and various annotations that make models easier to understand (such as comments or create lines), but do not add to the formal semantics of the models. Models transform shorthand notations from the concrete syntax of [ITU-T Z.103] into concrete syntax of [ITU-T Z.102] or [ITU-T Z.101].

7.4 Data and action language in SDL-2010 – [ITU-T Z.104]

[ITU-T Z.104] is the ITU-T Recommendation containing the part of the specification of SDL-2010 that adds the data and action language used to define data types and expressions. In SDL-2010 the use of different concrete data notations is allowed, such as the SDL-2000 data notation or C, with bindings to the abstract grammar and the predefined data package.

The underlying data model is fundamental to behaviour and provides sorts of data such as Boolean and Integer that are used in other language features. For that reason this underlying model and an overview of predefined data sorts and constructs is given in Annex D of this Recommendation.

[ITU-T Z.104] does not define general reference or object data types or creation of data items other than as variables that are part of an agent, procedure or state instance. These issues are covered by [ITU-T Z.107].

7.5 SDL-2010 combined with ASN.1 modules – [ITU-T Z.105]

[ITU-T Z.105] provides a mapping for ASN.1 modules to features defined in the rest of the Specification and Description Language recommendations for SDL-2010, so that the ASN.1 modules define data items that are used with the rest of SDL-2010.

7.6 Common Interchange Format for SDL-2010 – [ITU-T Z.106]

[ITU-T Z.106] provides alternative textual syntax for the graphical syntax items defined in [ITU-T Z.101] to [ITU-T Z.105] that is used as a Common Interchange Format (CIF) between SDL-2010 tools. The basic level of CIF provides only a textual equivalent of graphical items. The full CIF is intended for the interchange of graphical SDL-2010 specifications (SDL-GR) so that the drawings are recognizably the same.

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

7.7 Object-oriented data in SDL-2010 – [ITU-T Z.107]

[ITU-T Z.107] defines the object-oriented data features of the Specification and Description Language building on the foundation of the data definitions and expressions defined in [ITU-T Z.104].

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021

Annex A

Abstract syntax index

(This annex forms an integral part of this Recommendation.)

The abstract syntax index consists of the following table that lists the abstract grammar syntax rules of SDL-2010, where they are defined and redefined. Some abstract grammar syntax rules are defined in only in [ITU-T Z.102] or [ITU-T Z.104] as shown in the table below, in which case the column for [ITU-T Z.101] in the table below is shown blank. It is expected that users of the SDL-2010 Recommendations have access to machine-readable copies of the Recommendation texts and are therefore able to use computer software to locate the definitions and uses of the abstract grammar syntax rules. An abstract grammar syntax rule name should always be in italics, and the definition of a rule should start on a line in the original Microsoft Word text with the style "z.100 abs syntax 1st line".

Abstract syntax rule name	ITU-T Z.101	ITU-T Z.102	ITU-T Z.104	ITU-T Z.107
Abstract		definition		
Action-return-node	definition			
Activation-delay	definition			
Active-agents-expression	definition			
Active-expression	definition		redefinition	redefinition
Actual-parameters	definition			
Agent-definition	definition			
Agent-formal-parameter	definition			
Agent-identifier	definition			
Agent-instance			definition	
Agent-instance-pid-value			definition	
Agent-kind	definition			
Agent-name	definition			
Agent-qualifier	definition			
Agent-type-definition	definition	redefinition		
Agent-type-identifier	definition			
Agent-type-name	definition			
Agent-type-qualifier	definition			
Aggregation-kind	definition			redefinition
Alternative-expression	definition			
Any-decision		definition		
Any-expression			definition	
Argument	definition			
Assignment	definition			
Boolean-expression	definition			
Break-node		definition		
Call-node	definition	redefinition		

Documents
Document Sharing Hub
Full document details: ITU-T Z.100 2021