

Methods for Testing and Specification (MTS); Strategy for the use of formal SDL for descriptive purposes in ETSI products



European Telecommunications Standards Institute

Reference

DTR/MTS-00049 (a0c00ics.PDF)

Keywords

Methodology, SDL, protocol

ETSI Secretariat

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

X.400

c= fr; a=atlas; p=etsi; s=secretariat

Internet

secretariat@etsi.fr
<http://www.etsi.fr>

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

Contents

Intellectual Property Rights.....	4
Foreword	4
1 Scope.....	5
2 References.....	5
3 Definitions and abbreviations	5
3.1 Definitions	5
3.2 Abbreviations.....	6
4 Strategy for the use of SDL for descriptive purposes	6
4.1 Background.....	6
4.2 Basic requirements.....	7
4.2.1 Readability	7
4.2.2 Correlated SDL and text.....	8
4.2.3 Unambiguity	8
4.2.4 Syntactically and semantically correct	8
5 Overall strategy statement.....	9
Annex A (informative): Bibliography.....	10
History	11

Intellectual Property Rights

ETSI has not been informed of the existence of any Intellectual Property Right (IPR) which could be, or could become essential to the present document. However, pursuant to the ETSI Interim IPR Policy, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

1 Scope

The present document describes a strategy to be followed in the development of guidelines for the use of formal SDL for descriptive purposes within ETSI deliverables. The purpose of the present document is to identify the criteria which descriptive SDL should meet and to offer, in general rather than specific terms, methods and techniques which will be provided in an ETSI Guide planned for publication in September 1998. The guidelines being elaborated under Work Item DEG/MTS-00050 will assist rapporteurs of standards specifying behaviour to produce formal SDL that can be easily read and understood by non-experts in SDL.

Guidance on whether SDL specifications should be normative or non-normative is outside the scope of this strategy.

The strategy described in the present document applies to all standards that specify behaviour. Examples of these are protocols and services. The present document complements rather than supplants the methodologies described in ETS 300 414 [1], ETR 298 [2] and EG 201 015 [3].

2 References

References may be made to:

- a) specific versions of publications (identified by date of publication, edition number, version number, etc.), in which case, subsequent revisions to the referenced document do not apply; or
- b) all versions up to and including the identified version (identified by "up to and including" before the version identity); or
- c) all versions subsequent to and including the identified version (identified by "onwards" following the version identity); or
- d) publications without mention of a specific version, in which case the latest version applies.

A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

- [1] ETS 300 414 (1995): "Methods for Testing and Specification (MTS); Use of SDL in European Telecommunication Standards; Rules for testability and facilitating validation".
- [2] ETR 298 (1996): "Methods for Testing and Specification (MTS); Specification of protocols and services; Handbook for SDL, ASN.1 and MSC development".
- [3] EG 201 015 (1997): "Methods for Testing and Specification (MTS); Specification of protocols and Services; Validation methodology for standards using SDL; Handbook".
- [4] ITU-T Recommendation Z.100 (1993): "Specification and Description Language (SDL)".
- [5] ITU-T Recommendation Z.100 Addendum 1 (1996): "Corrections to Recommendation Z.100, CCITT Specification and Description Language (SDL)".
- [6] ITU-T Recommendation Z.105 (1994): "SDL combined with ASN.1".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following definitions apply:

current SDL: SDL as defined in ITU-T Recommendation Z.100 (1993) [4] in combination with ITU-T Recommendation Z.100 Addendum 1 [5].

formal SDL: Specifications using SDL that is syntactically correct and conforms to the static semantic rules but which is not necessarily complete or executable.

informal SDL: Specifications using SDL that does not conform to the established rules of syntax or semantics.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation No. 1
MSC	Message Sequence Chart
SDL	Specification and Description Language

4 Strategy for the use of SDL for descriptive purposes

4.1 Background

In general, the purpose of a standard and, hence, the formal descriptions contained in the standard, is to specify the minimum requirements that shall be met in order to guarantee interworking and interoperability of networks and equipment. The use of SDL in international standards has been accepted for many years but the "quality" of SDL produced has varied considerably from standard to standard and, in most cases, can only be regarded as informal illustration. In recent years, some technical bodies have begun to use SDL in a more formal way, producing complete executable models which can be simulated and validated. The Core INAP CS2 specification produced by STC-SPS3 is an excellent example. However, the production of a validation model requires a knowledge of the language and skills which are not possessed by many of the rapporteurs of protocol standards. Also, a fully executable SDL specification of a protocol or service can be very difficult to read without those same skills. There is, therefore, a need to find a "middle-ground" where the SDL used in standards is formally expressed but reasonably easy to write and read without significant training in the language. It must also be possible for somebody more skilled in the use of SDL to produce an accurate validation model without ambiguity.

The objectives of the planned guidelines for the production of descriptive SDL are to improve the readability, usability and understandability of the SDL. However, it should be recognized that the descriptive SDL is unlikely to be the same as that which exists in a corresponding validation model even though it would be possible to develop a validation model from the descriptive SDL and, conversely, to derive descriptive SDL from a validation model. Figure 1 illustrates the relationship between these two levels of SDL specification.

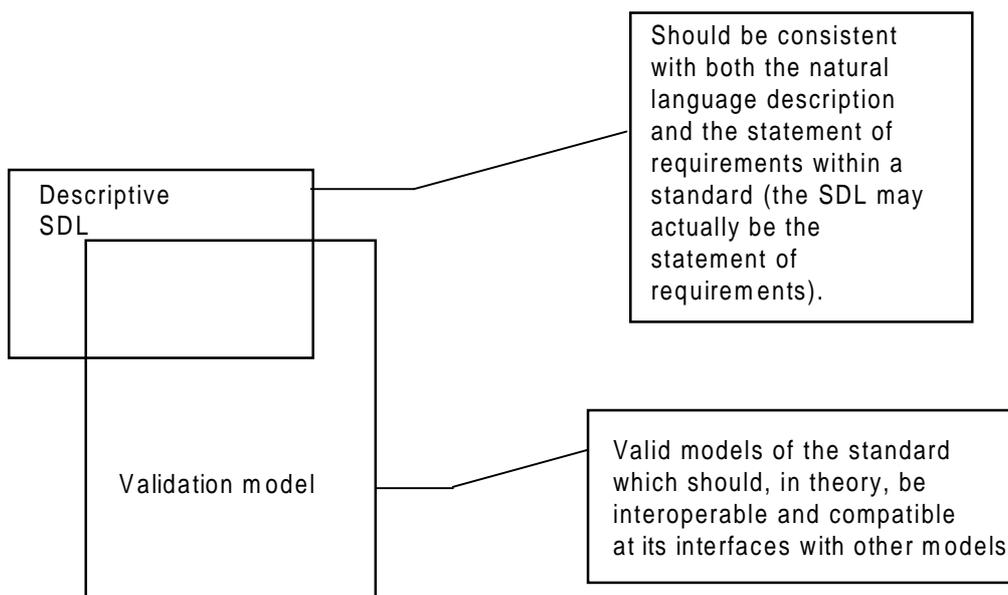


Figure 1: SDL models

The development of models using formal languages such as SDL is very similar to the process of software engineering. The quality of the SDL description can benefit from the adoption of some of the approaches used in software engineering even though the model produced is unlikely to be executable without modification and elaboration.

4.2 Basic requirements

Before guidelines for the use of formal SDL as a descriptive tool can be written, it is essential that the required characteristics of such SDL are established and understood. Within the standards-writing community, there is a growing body of opinion that advocates the use of SDL as the normative specification of behaviour within a standard with text supporting and clarifying the requirements expressed in the SDL. It must, therefore, be possible for descriptive SDL to be used not only as an informative specification supporting normative text but also as a normative specification supported by additional text.

If descriptive SDL is to be of consistent value across a wide range of protocol and service standards, it must meet the following criteria:

- it must be readable. That is to say that it must accurately describe the required behaviour in a way that is understandable by readers possessing only a minimum knowledge of SDL;
- it must provide a clear correlation between the SDL and the associated text specification;
- it must be unambiguous. In addition to the requirements for readability, it must contain sufficient, clearly-expressed information to enable an accurate validation model to be derived from it;
- it must be syntactically correct and, as far as possible, comply with the SDL rules for static semantics.

There are other characteristics that have been considered in the specification of the guidelines and these include maintainability and ease of development into a more complex model for validation or implementation. While these are important and are reflected in the guidelines, they have been considered to be secondary to the four characteristics identified above.

4.2.1 Readability

Software engineering techniques that can be applied to improve readability are:

- layering (information hiding);
- naming;
- structuring;
- documenting.

Layering of information in any description aids gradual understanding by presenting that information at successive levels of detail. The identification of what needs to be done is separated from the specification of how it is done. It aids validation and identification of test cases and provides a framework for development, so that a developer can concentrate on one layer at a time. On the other hand, too much layering is not good for readability, because a reader has to jump from one description to another in order to grasp the overall meaning. For this reason, functions that logically belong to one layer should not be further partitioned for "programming" reasons.

Naming of different entities is extremely important for readability. For example, the names chosen should be memorable and carry as much meaning as possible. Also, procedure names should be chosen so that they are descriptive of *what* the procedure does. In this way the calling diagram (or text) can be read at an abstract level without examining the procedure to find out *how* it is defined in detail.

SDL supports constructs which enable clear descriptions of structure to be produced (systems and blocks). Even so, readability can suffer if a system, for example, is decomposed into numerous constituent blocks without introducing intermediate groupings. Behaviour descriptions (processes) can be structured in several different ways, some of which are more readable than others. With appropriate guidance, readability can be significantly improved by the careful construction of the specification. For example, the processing of error conditions in a protocol specification can often result in complex and unreadable decision trees. Techniques are available which help to avoid such problems.

SDL descriptions can be supplemented with additional information in the form of documentary comments or references to relevant information in textual parts of the standards. These could include an overall informal textual description of the whole SDL model and its constituent parts (blocks, processes and procedures) and comments attached to tasks, inputs and outputs. These comments could, for example, help the reader to understand sources or destinations of signals or some aspects of parameter handling.

4.2.2 Correlated SDL and text

As the purpose of descriptive SDL is to complement a text-based standard with additional detail and clarity it is important to make the relationship between the relevant text and the corresponding SDL as clear and as obvious as possible. In fact, this relationship can have implications on the way that the textual standard is structured. In the case where text already exists, it will be more difficult to show this relationship cleanly. Experience in applying SDL (or indeed any similar software engineering technique) indicates that in many cases a standard could benefit from a cleaner, more consistent structure (both at high levels and low levels of detail).

It would, therefore, be clearly beneficial to provide simple ways of achieving this without adding undue overhead to the SDL writer but which give real benefits to the reader.

Other ways to show a clearer correlation are by the consistent use of sensible and relevant naming conventions and by providing (for example, as comments) references to clauses from SDL diagrams. In the latter case care must be taken to avoid this becoming a burden for the specifiers, especially considering the possible maintenance aspects. In the future these references could even be supported by tools providing hypertext links between the text and the SDL. Although this would be very useful for the user of standards, it is unlikely that the average rapporteur would have the time to perform this task without good tool support.

While not suggesting that a textual standard should be structured simply to follow the SDL, the guidelines encourage the development of the text and SDL in tandem, to the mutual benefit of both descriptions.

4.2.3 Unambiguity

It is essential that any formal specification of behaviour is expressed in a way that cannot be misunderstood by its readers. This applies equally to the case where the reader is interested only in gaining a general understanding of the specified behaviour as to the case where an executable model is being developed from the descriptive SDL. Ambiguity can often arise, for example, from:

- the misuse of convenient shorthand notations;
- poor identification of the source of inputs and the destination of outputs;
- insufficient identification of the function of a procedure;
- unnecessarily complex process structures.

The guidelines offer alternative methods for avoiding such ambiguity in descriptive SDL.

4.2.4 Syntactically and semantically correct

Although SDL produced for descriptive purposes need not be a complete specification of a system, it is essential that it follows the rules of syntax and static semantics established for the language. This helps to avoid ambiguity and should improve readability. In addition, it will greatly simplify the task of developing a validation model or even an implementation model from the descriptive SDL. The current SDL coupled with ITU-T Recommendation Z.105 [6] allows data structures to be specified using ASN.1 - the language of choice for most protocol standards rapporteurs for specifying message structures - and even permit process diagrams to stand alone as valid SDL without their associated system and block specifications.

The guidelines offer practical directions to ensure that the syntax and static semantics of SDL are respected by rapporteurs using the language.

5 Overall strategy statement

The guidelines which will be the result of Work Item DEG/MTS-00050 and planned for publication in September 1998 are intended for rapporteurs using SDL to describe stimulus-response behaviour in ETSI deliverables such as protocol specifications. In the past, such descriptions either used SDL that does not completely conform to the ITU-T Recommendation Z.100 [4] standard (that is, the description are not "formal SDL") or were written in natural language only.

The result has been that additional effort is required to understand the ETSI deliverable, to produce conformance or other tests from it and to specify and produce implementations. The objective in developing the guidelines is to provide a means of producing formal SDL specifications which can be understood by readers who are not experts in the language. Application of the guidelines should benefit the producer and users of the ETSI deliverable.

The use of SDL as a descriptive tool is constrained by applicable ETSI deliverables and ITU recommendations, ETS 300 414 [1], ETR 298 [2], EG 201 015 [3] and ITU-T Recommendations Z.100 [4] and Z.105 [6] with consideration also given to object orientation, current SDL and evolving MTS deliverables. The strategy used in developing the guidelines has been to establish criteria which should be met by descriptive SDL specifications (subclause 4.2) and then to investigate the use of SDL in existing standards to determine to what extent advice is needed to enable rapporteurs to produce, with minimal effort, formal SDL which meets those criteria. The result of this is a set of guidelines which:

- suggest ways in which the SDL can be written to aid readability through the use of software engineering techniques (subclause 4.2.1);
- encourage the development of the text and SDL in tandem, to the mutual benefit of both descriptions, though not suggesting that a textual standard should be structured simply to follow the SDL (subclause 4.2.2);
- offer alternative methods for avoiding ambiguity in descriptive SDL (subclause 4.2.3);
- offer practical directions to ensure conformance to the syntax and static semantics of SDL so that maximum benefit can be derived from the language and supporting tools (subclause 4.2.4).

The guidelines are intended to be easy to use without the need for significant experience in SDL itself. Extensive use is made of examples to show formal SDL as an alternative to non-conformant or undesirable descriptions.

Annex A (informative): Bibliography

The following material, though not specifically referenced in the body of the present document, gives supporting information:

- ITU-T Recommendation Z.120 (1993): "Message Sequence Chart".

History

Document history		
V1.1.1	June 1997	Publication