ETSI
TECHNICAL
REPORT

ETR 184

April 1995

Source: ETSI TC-MTS

Reference: DTR/MTS-00001

ICS: 33.020, 33.040.40

Key words: Methodology, Validation, SDL

Methods for Testing and Specification (MTS); Overview of validation techniques for European Telecommunication Standards (ETSs) containing SDL

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

New presentation - see History box

Whilst every care has been taken in the preparation and publication of this document, errors in content, typographical or otherwise, may occur. If you have comments concerning its accuracy, please write to "ETSI Editing and Committee Support Dept." at the address shown on the title page.

Contents

1 Scope

The purpose of this ETR is to identify future standardization work items necessary to support the implementation of a formal validation process into the production of all European Telecommunication Standards (ETSs) having a functional content expressed using the Specification and Description Language (SDL), defined in ITU-T Recommendation Z.100 [11].

NOTE: SDL is not the only language that can be used for formal specifications of function and behaviour. It is, however, the language currently recommended for ETSs.

The relationship of the validation process is considered within the context of ETS and test suite development. The constituent parts of the validation process are identified and described and the capabilities of existing modelling tools are reviewed against ETSI's requirements. It is likely to take a number of years before an ideal validation procedure interleaved with the ETS development process can be realised and so some short term measures are proposed. Finally, a work plan for the progression of formal validation methods within ETSI during 1995 and 1996 is presented.

2 References

For the purposes of this ETR, the following references apply:

[1] Working Procedures for the Technical Assembly and its Working Bodies for the European Telecommunications Standards Institute (TAWP) - Edition 2 (March 1994).

[2] Statutes and Rules of Procedure of the European Telecommunications Standards Institute.

[3] CEN/CENELEC Internal Regulations: Part 3 (1991): "Rules for the drafting and presentation of European standards (PNE Rules)".

[4] Manual for the application of the ETSI style sheet for documents using Microsoft Word for Windows 2.x (1993).

[5] ETS 300 406 (1995): "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications Standardization Methodology".

[6] prETS 300 414 (1994): "Methods for Testing and Specification (MTS); Use of SDL in European Telecommunication Standards; Rules for testability and facilitating validation".

[7] ETR 102 (1993): "European digital cellular telecommunications system (Phase 2); Technical performance objectives (GSM 03.05)".

[8] DTR/MTS-00013: "Methodology for the specification of ETSI protocols and services when using SDL".

NOTE: This work item is still under development and the resulting ETR is expected to be published at the end of 1995.

[9] ITU-T Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".

[10] ITU-T Recommendation Z.100 (1988): "Specification and Description Language (SDL)".

[11] ITU-T Recommendation Z.100 (1992): "Specification and Description Language (SDL)".

[12] ITU-T Recommendation Z.110 (1988): "Criteria for the use and applicability of formal Design Techniques".

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

[14] ISO/IEC 9646-2 (1991): "Information technology - OSI conformance testing methodology and framework - Part 2: Abstract test suite specification".

[15] ISO/IEC 9646-3 (1991): "Information technology - OSI conformance testing methodology and framework - Part 3: Tree and Tabular Combined Notation".

[16] Holzmann, Gerard J.: "Design and Validation of Computer Protocols", Prentice-Hall (1991)".

[17] West, Colin H.: "Computer Networks and ISDN Systems 24 219-242: Protocol validation - principles and applications", North-Holland (1992).

[18] West, Colin H.: "FORTE'93, Boston: The Challenges Facing Formal Description Techniques", North-Holland, Amsterdam, (1994).

[19] Hogrefe, D.: "Third SDL Forum, The Hague: Simulation of large SDL systems", Elsevier Science Publisher, Amsterdam, (1987).

[20] Håkansson M., Persson Ö.: "Performance Simulation for SDT", Dept. of Communication Systems, Lund Institute of Technology, Lund, Sweden (1994) Master thesis.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETR, the following definitions apply:

validation: the process by which appropriate methods, procedures and tools are used to evaluate that a standard:

- satisfies the purpose expressed in the record of requirements on which the standard is based;
- can be fully implemented;
- when implemented is able to offer all the functionality and performance expressed in the record of requirements on which the standard is based;
- conforms to the established criteria for standards.

aAssertion: a statement of system properties that can be evaluated as being true or false.

behaviour tree: a description of the behaviour of a system in the form of a tree. Nodes of the tree are different states of the system, with the root node representing the initial system state. Branches between nodes represent transitions between states caused by external or internal stimuli.

deadlock: a state of a system such that no progress is possible.

formal model: a system model that is expressed in a language such that its behaviour can be unambiguously interpreted.

invariant: a property of a system that is true in all states of the system.

semantics: the meaning of a specification that relates it to the real system it defines.

syntax: the form of a specification, the rules specifying the use of the elements of the specification language.

3.2 Abbreviations

For the purposes of this ETR, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
ICS	Implementation Conformance Statement
ISDN	Integrated Services Digital Network
MSC	Message Sequence Chart
OSI	Open System Interconnection
PICS	Profile Implementation Conformance Statement
PNE Rules	Rules for the drafting and presentation of European Standards (Présentation de Normes Européenes)
PTN	Private Telecommunication Network
SDL	Specification and Description Language
TTCN	Tree and Tabulation Combined Notation

4 A framework for validation

4.1 The benefits of validation

During the last ten years, validation technology has developed from being a research topic to a practical method of finding errors in complex distributed systems, that is now available as part of several, commercially available software development tools. The formal validation of system specifications is accepted by equipment manufacturers as a significant means of avoiding the costs of design errors.

Experience has shown that validation is capable of finding most errors in executable system specifications that contain in excess of ten thousand lines of code. Once a system has been validated, the numbers of errors found during product development is significantly reduced. It is common for validation to find of the order of one error per 100 lines of executable system specification.

Validation has been used to improve the quality of international standards. A paper presented at the Third SDL Forum in The Hague [19] describes how the CCITT SS 7 protocol has been intensively simulated in order to check its validity. A number of errors were found and reported back to the respective CCITT Study Group. Another example is the Open Systems Interconnect, OSI, Transaction Processing protocol that has been validated by non-exhaustive validation. Details of this work can be found in a paper presented to the FORTE'93 conference in Boston [18] but, in summary, over 200 errors were found in the standard and resulted in substantial changes in the developing standard. The study took 18 man months and 50 different versions of the standard were validated. Both of these studies significantly improved the quality of the standards.

Similar benefits can be expected when validation is applied to ETSI standards. It is difficult to determine the general quality of published ETSs as problem reports have not been analysed in the past. Also, many ETSs are written by volunteer rapporteurs and, thus, the cost of rework is borne by member organisations rather than the Costed Work Programme. The following statements can, however, be made:

- the presentation style of SDL diagrams, the level of detail in them and the use of Message Sequence Charts (MSCs) and Abstract Syntax Notation One (ASN.1), is inconsistent across the existing range of published ETSs. As a result, the interpretation of standards by implementers is variable;

- almost all the checking of syntax and internal consistency of ETSs containing SDL diagrams is carried out by inspection rather than automatically. Manual methods cannot be expected to find complex specification errors;

- some ETSs have failed their Public Enquiry stage as a result of specification errors that could have been found by formal validation. Given that the Public Enquiry lasts for approximately four months, such a failure can easily delay publication by a year.

The incorporation of formal validation techniques into the ETS development process will help to overcome all of these problems.

4.2 The requirement for validation of ETSs

An ETS is produced within the context of the TA Working Procedures [1] that give additional information to the ETSI Statutes and Rules [20] with an ETS being identified as a specific type of ETSI deliverable. **Section B of the Working Procedures clearly highlights planned, formal validation as a requirement for all ETSs** (clauses B.5.1 and B.6.3.1).

The extent of validation depends on the technical nature of the standard. For example, an ETS comprising a glossary of terms will require that only format, style and completeness are validated whereas a protocol specification would additionally need the use of modelling techniques to validate the formal specification of the protocol. Those standards that can be implemented directly within a telecommunications product, also require a conformance testing specification to be published. Determining the consistency between a product standard and the corresponding test specification is part of the validation of both standards.

Validation is an important part of the quality assurance of deliverables. A Technical Committee (TC) chairman is responsible for all aspects of quality management within the TC and the TC responsible for preparing and approving an ETS is also responsible for providing a supporting statement regarding the validation requirements for the ETS. This may include:

- the amount and type of validation necessary;
- the expected validation results;
- a validation plan.

This responsibility may often be partially delegated to a Sub-Technical Committee (STC). In this ETR, reference to a TC means the TC or any body the TC has delegated to undertake the work.

It is necessary to determine the exact validation requirements that need to be specified before the development of an ETS can begin.

4.3 International validation activities

Other organisations have ongoing activities that are intended to improve the quality of their standards. It is important for ETSI to plan validation activities that are compatible with similar work at the international level. The two major organisations outside ETSI that are active in this field are ITU-T and ISO.

ITU-T Study Group 10 is responsible for the studies relating to technical languages and methods for telecommunication applications. The program of work of this group includes the subject area, "Testing Based On Formal Specifications And Verification Of Formal Specifications".

Two work items in this area are "Correctness Of Formal Specifications" and "Formal Methods In Conformance Testing".

The former, started in 1993, is expected to be completed in 1996, and is under the sole responsibility of ITU-T.

The study group is working jointly on the latter document with ISO which has defined a similar project, ISO/IEC JTC1 SC21 P54. It should reach Committee Draft status at the end of 1994, and it is planned to publish an International Standard in 1996.

4.4 The elements of an ETS

To avoid ambiguity, an ETS specifying behaviour should use a formal modelling technique to express that behaviour. The modelling method recommended in ETS 300 414 [6] is SDL.

Such an ETS should contain:

- text specifying the standardised behaviour in unstructured prose;
- Message Sequence Charts specifying the flow of information between blocks of the standardised system and with the environment;
- SDL specifying the behaviour in structured and graphical terms;
- ASN.1 code specifying the encoding of messages.

An ETS will also identify in its normative references those other standards with which it needs to be consistent. Conformance tests for a standard are normally specified in a separate ETS using Tree and Tabulation Combined Notation (TTCN) as defined in ISO/IEC 9646-3 [15].

The combination of SDL, MSCs and ASN.1, described respectively in ITU-T Recommendations Z.100 [11], Z.120 [13] and X.208 [9], produces a definition of the required behaviour that is precise and which can be validated with automatic tools. Unstructured prose can only be validated by inspection which is prone to inaccuracy. The validation of standards expressed in a formal specification, such as SDL, can be far more extensive and accurate. However, in most ETSs the SDL is considered to be informative with the supporting prose being normative. ITU-T Recommendation Z.110 [12] identifies the fact that the full benefits of validation can only be realised if the prose and the formal specification are given equal status.

4.5 Validation of an ETS against ETSI rules

All ETSs are required to conform to language and form requirements specified in the PNE Rules [3] and the ETSI style rules [4]. Tools are already in widespread use in the development of draft ETSs to ensure that some of these fixed requirements are universally applied. Other rules are validated by visual inspection. Validation of ETSs can be simplified if conformance to the rules of form and language is enforced during the development process. Tools for this purpose might include:

- a grammar checker based on the PNE Rules [3];
- a document template for each type of ETS;
- abbreviation identification tools.

5 Validation of a formal model

A formal model of a system is one that is expressed in a precisely defined, formal language. Such a model has two significant advantages:

- it represents a precise, unambiguous specification;
- it provides a basis for analysing the system properties, particularly when the analysis can be automated.

5.1 Analysis of a formal model

There are many techniques that can be used to analyse a formal model. Static analysis techniques are those that do not require execution of the model. Dynamic execution techniques include both simulation and testing, as well as specialised validation techniques, all of which depend on the ability to execute the model. These are described in the remainder of this clause, together with some informal analysis techniques and other techniques that are the subject of on-going research.

5.1.1 Static analysis

Static analysis is confined to evaluating properties of the system without executing the functions specified. It can be divided into activities, syntax analysis and semantic analysis. Syntax analysis checks that the rules associated with the use of the language and with the definition of the system are satisfied. Semantic analysis checks that the system definition is well defined, complete and self consistent. This includes verifying that types and parameters are consistently used, and that the definition contains no references to undefined system components and includes no unreferenced system components.

Static analysis can demonstrate that the form of a model is correct, but not that the functions it describes are correctly specified. A model that has been analysed in this way is equivalent to a program that has been successfully compiled. Further analysis is required to demonstrate that it correctly expresses the intended functions.

5.1.2 Dynamic analysis

Dynamic analysis considers the behaviour of the model when executed. This is a more complex task but can address two classes of system properties.

The first covers the dynamic semantics of the model, namely whether or not the functions defined in the model correspond to those defined by the system requirements. Such an analysis requires that the model is executed. During execution, its behaviour is evaluated with respect to assertions and invariants that encapsulate the desired system behaviour. These assertions and invariants are generally particular to the model being analysed.

The second class of system properties includes the general properties that can be expressed in terms of assertions that are common to broad classes of systems and so can be evaluated in a highly automated way. Examples of such properties are freedom from deadlock and the use of variables before they are defined.

A range of dynamic analysis techniques can be defined, and classified in many ways. The classification given in subclauses 5.1.2.1 to 5.1.2.4 is somewhat arbitrary.

5.1.2.1 Testing the formal model

In many cases the formal model can be expressed as an executable program. This can be checked by testing in the same way as any other program is tested. Usually, test scenarios will have to be developed. A suite of test cases is prepared, each one designed to exercise particular functions of the model. When applied to the model, the observed behaviour of the model is analysed to see if it corresponds to the expected outcome. In most cases, however, testing does not achieve complete coverage of the behaviour.

The test scenarios may be described using MSCs and the consistency between these and the SDL can be validated.

The technique used is to build a set of traces corresponding to the behaviour tree or at least those traces that are relevant for the particular MSC being verified (guided partial search). Each of the traces generated corresponds to a possible execution path.

To be consistent with the MSC, a trace should:

- contain all events defined in the MSC;
- involve only process instances which are part of the MSC;
- comply with the partial ordering defined by the MSC.

The MSC is considered to be valid if a trace corresponding to these criteria is found.

5.1.2.2 Exhaustive validation

It is necessary to distinguish between two classes of validation methodologies, exhaustive and non-exhaustive. The first contains terminating algorithms that can be applied to a specification. When executed, they perform a complete analysis of a system to determine its properties. Such algorithms can be used to determine, for example, whether or not a system is deadlock free. Non-exhaustive algorithms may not terminate, so that they cannot be used to make precise statements about system properties, they can however be used to search for errors. The most significant exhaustive validation methodology, because of its power, simplicity and the ease with which it can be automated is reachability analysis. The aim of reachability analysis is to compute the graph of all system states that can be reached from its initial state. Each state of the system is analysed to determine all of the transitions from it that can be exercised and to check that it conforms with predefined assertions and invariants. All states that are reached when the transitions are exercised are compared with those already present in the reachability graph so that each reachable state is analysed only once. The analysis of the reachability graph is equivalent to an analysis of all sequences that the system can execute, yet is considerably more efficient. If all reachable states of the system can be determined, properties such as freedom from deadlock can be proven.

The following list contains examples of the system properties that can be analysed by a reachability analysis tool. A definitive list is generally a function of the language used to define the system:

- detection of unexercised functions;
- deadlock;
- implicit signal consumption;
- unsuccessful create;
- output errors (for example, *output with no receiver*);
- data operator error (for example, *division by 0*);
- sub-range violation;

- index out of range;
- no path through decision;
- violations of Invariants and Assertions.

Although more powerful than testing, reachability analysis is not universally applicable. A complete analysis of a complex protocol may require the computation of a reachability graph that contains an excessive number of states (the so-called *state explosion* problem).

5.1.2.3 Non-exhaustive validation

When a system is sufficiently complex that reachability analysis is not applicable, non-exhaustive validation techniques need to be used. Experience with reachability analysis tools has shown what type of validation can be usefully applied to complex systems. A reachability analysis will generally report instances of a single error in many different states of the system. In general, only one instance needs be analysed to understand and correct the error. Errors are reported many times because they are simple in comparison with the system itself. That is to say, they can be described in terms of a subset of the parameters that collectively define the system behaviour.

These observations have suggested that an effective way to find errors in complex systems would be to analyse a random subset of the reachable states, see "Computer Networks and ISDN Systems" [17]. This is equivalent to a reachability analysis where, instead of exercising all transitions from a system, only one is exercised at random.

The disadvantage of the method is that there is no termination and it is thus not possible to demonstrate that a system is error-free. Application of the method has however demonstrated that it is an extremely effective way of finding errors in systems that are too complex for reachability analysis.

5.1.2.4 Simulation

Several of the validation methodologies discussed in subclause 5.1 can be viewed as some form of systems simulation. They all, however, address the analysis of the correctness of system functions and do not address the evaluation of system performance. For completeness, a brief description of those analyses that evaluate system performance is included here. Performance analysis evaluates system properties such as delay, throughput, and other timing related properties. A formal, functional model is a useful starting point for such an analysis, but needs to be supplemented with timing and performance data.

5.1.3 Applying informal validation techniques to formal model

In practice, informal techniques have proved very useful in checking the correctness of a formal model. The most common techniques are walk through and inspection with a check-list.

In a walk through session, the author of a specification and a number of other experts who have not participated directly in its development, go through the document step by step. Problems discovered are not solved during the session. The author is expected to defend his work while remaining open to alternative suggestions. The success of such a session depends on the climate and the will to co-operate. It should not be used to evaluate the author's capabilities. Higher management should not, therefore, participate.

Inspection is usually carried out by trained inspectors who go through a specification according to a check-list. The author of the specification may or may not participate. Such a check list depends much on the particular specification technique and capability of tools used. For SDL it could contain questions such as:

- are there any signals received by a process that never appear in an input?
- do the answers to the question in a decision cover every value of the question?

5.1.4 Other validation techniques

Research is ongoing in various validation techniques other than those described in this ETR. An example of this research is Holzmann's "Design and Validation of Computer Protocols" [16]. A number of similar projects show promise but are unlikely to be available in the form of tools in the foreseeable future.

One technique that is well established in academic environments has been developed in the context of process algebras and can be used for any specification mechanism that defines a behaviour tree. Here equivalence between two behaviour specifications according to a predefined equivalence relation is to be proven. As an example, consider an OSI service specification and the OSI protocol specification that is supposed to supply the service. If the service specification is known to be valid, an equivalence proof with the protocol specification will show validity also for the protocol specification.

In practice this kind of proof is most likely to demonstrate problems in both specifications, particularly if neither of the two specifications have been exhaustively validated previously.

The practical value of rigorous proofs is not clear and it is uncertain that tool support will be available in the near future. To meet immediate practical needs, simulation can be used to check partial equivalence.

5.2 SDL - specific requirements for validation tools

In order to be able to assess the capabilities of a tool for validating SDL-based standards, it is necessary to review the requirements for such a tool. These requirements encompass the spectrum of activities discussed in subclause 5.1. Subclauses 5.2.1 to 5.2.3 describe how some of the activities have to be tailored to match SDL and other specification techniques used in ETSs.

5.2.1 Editors, syntax and static semantics checking

Since SDL is used in combination with MSC, ASN.1 and TTCN, editors for these languages should be provided with or supported by any SDL tool.

It is clearly important that tools that can check the syntax of SDL specifications should be available. Syntax checking facilities are also required for ASN.1, MSCs and TTCN when the latter are also incorporated in an ETS.

Facilities are required for checking the static semantics of SDL and of the other formal constructs mentioned in the previous section.

5.2.2 Testing facilities

Once an SDL system has been defined and demonstrated to be syntactically correct it has to be possible to subject it to dynamic analysis and to observe the behaviour when external inputs are applied. The purpose may be either to debug individual functions within one process or to apply complete test sequences for analysing the behaviour of interacting system components. It is highly desirable that this latter function is such that test sequences can be expressed in the form of MSCs and/or TTCN. It is then possible not only to test functions defined in SDL, but also to check the consistency of SDL and MSCs or TTCN definitions.

5.2.3 Reachability analysis

An SDL system usually consists of a number of communicating processes. Some tools can only perform a reachability analysis for a single process. This is not sufficient. In order to use reachability analysis, the tool has to map the states of the individual processes of an SDL system and its supporting communications layers to a single state space. Appropriate means to limit the size of the state space are helpful.

5.2.4 Tools supporting validation

There are two basic families of tools that are used to validate formal models, simulators and validators. The boundary between these two is not well defined but it is convenient to make a distinction for the purpose of assessing the tools.

5.2.4.1 Simulators

Simulators are used to implement the testing of the formal model discussed in Subclause 5.1.2. They enable a system model to be executed so that the user can monitor its response to sequences of defined input signals provided either interactively or in a batch mode. Such tools should support:

- inputs specified as lines of text;
- inputs specified in the form of MSCs;
- output reports in text or MSCs;
- telecommunications standards that specify only partial systems. For these it is necessary that the simulation model can communicate with other previously validated models or with external applications.

5.2.4.2 Validators

Validators use the reachability methods to search for all (or most) possible states that can exist within a formal model. Such tools should support:

- validation of MSCs;
- reachability analysis;
- the description of correctness requirements in terms of:
    - assertions;
    - system invariants;
    - deadlocks.

5.2.5 Summary of requirements

A rapporteur for an ETS is expected to provide a standard environment for the task. This environment is an IBM compatible PC running Microsoft Word-for-Windows. It is not reasonable to expect that the whole range of analysis and simulation requirements could be met by such a platform. However, those parts of an SDL modelling tool that help to ensure completeness and syntactical correctness should be available to all rapporteurs so that time is not wasted trying to model an invalid specification. Thus, the requirements are divided into two groups; those which apply specifically to whatever platform is specified for rapporteurs (table 1) and those which could be made available as a central validation service (table 2).

Table 1: Summary of requirements for a SDL support tool for rapporteurs

Feature	Requirement
Graphical editor for SDL and MSC	Essential
Syntax analysis	Essential
File interchange facilities with simulators and validators	Essential
Language sensitive ASN.1 editor	Desirable
Consistency analysis between all parts of the specification (MSC, SDL and ASN.1)	Desirable

Table 2: Summary of requirements for a SDL modelling and analysis tool

Feature	Requirement
All the features of the rapporteurs' tool	Essential
Static and dynamic semantic analyser	Essential
Validation tool for MSCs	Essential
Exhaustive and/or non-exhaustive validation tools	Essential
Interactive simulation tool	Desirable
Simulation input scenario logging	Desirable
Acceptance of TTCN as input to analysis packages	Desirable
Graphical symbolic debugger	Desirable
Performance simulator	Desirable

6 The validation process

6.1 The validation model

Most validation activities require that the functions defined in an ETS be executed in some way. In general, the ETS itself will not be directly executable so its execution will require that a validation model corresponding to the ETS is developed. The nature and complexity of the model will depend on the ETS to be validated and the type of validation that is to be performed. In subclauses 6.1.1 to 6.1.5 the properties of the validation model and the differences between the model and the ETS that it represents are discussed.

6.1.1 Scope of the validation model

An ETS defines a set of processes and functions that collectively describe a particular standard. It does not completely define the environment within which they operate. The operating environment consists of, for example:

- functions provided by other standards referenced by an ETS;
- networks or other transport services that an ETS uses;
- systems that themselves use the functions defined by the ETS.

The validation model needs to take into account all components of the operating environment that are required to exercise the functions of the ETS that are to be validated.

Interactions with other elements of the operating environment may be modelled. For example, use of a transport service may be included by modelling the interactions with the transport service without including all of the internal functions needed to support the service. If, however, the transport service is defined by another ETS, it may be expedient to include much of the validation model developed for the latter, rather than develop a separate, simpler model of the service it provides. The decision on this also depends on the complexity that the validation model will have. Maintaining a library of validated models and components would be very useful in this respect.

6.1.2 Limiting the model

The validation model should include all of the functions defined by the ETS. The simplest way to do this is to use the SDL definition included in the ETS. However, in some cases it may be necessary to modify the definition because it is not possible to validate it in its most general form. Such modifications need to be defined on a case by case basis. A simple example will illustrate the type of modification that may be necessary.

EXAMPLE: If a protocol references message sequence numbers, it provides for recycling the sequence numbers when the limit of the sequence number field is reached. This limit may be a large number so that exercising the recycling functions may be difficult. For validation purposes, this limit may be set at an arbitrary small number so that it can be exercised with shorter message sequences.

Another example may be to choose specific options that the ETS offers to an implementer in order to reduce the size of the model.

6.1.3 Configuring the model

In general the Information Conformance Statement, ICS, proforma that comes with an ETS will have to be filled with a particular ICS in order to exercise a particular configuration. It is desirable that the validation model be formalised in such a way that the options are clearly identifiable. There should be a mapping from the ICS to the validation model. The validation should be performed by selecting different appropriate combinations of options.

If validating a profile, the combination of options is precisely defined in the profile definition. If validating a base standard, the combination of options could correspond to those defined in the profiles available at the time of the validation.

NOTE: The term ICS is the generalised abbreviation for "Implementation Conformance Statement". For Protocol ICSs, the abbreviation "PICS" is more commonly used.

6.1.4 Modelling unspecified functions

Most standards refer to at least some functions that need to be performed without specifying the details of how they are implemented. For example, a network may specify the functions of a name server, the protocol by which its services are accessed, the nature of the names and addresses it maintains, without specifying the way in which it stores and manipulates the data it maintains. An executable model nees to expand such functions to an executable form if they are exercised as part of the model.

6.1.5 Checking using the validation model

Since the validation model is a model of the system specified in the ETS, two types of errors may be discovered when investigating the model:

- errors due to the modelling, i.e. the system specified in the ETS has not been transformed correctly into the validation model;
- errors in the system specified in the ETS.

6.1.5.1 Checking against classified requirements

The classified requirements define the purpose and the basic requirements of the system. The validation model has to be checked against these in order to validate that the correct system has been specified.

If the classified requirements can be formalised, this can be done by experiment. Possible forms of formalizations are MSCs, assertions and invariants. Simulation can be used to check the compatibility of the MSCs with the validation model. Rigorous methods may be necessary to validate invariants or assertions. Currently there are no tools available to support such methods.

Other classified requirements that cannot be formalised can be checked by inspection, possibly using check lists.

If errors are found in the validation model, they have to be reinterpreted with respect to the ETS.

6.1.5.2 Validation of internal consistency

A number of checks can be performed on the validation model according to techniques defined in subclause 5.1.

6.2 Validation of performance

Performance validation of an ETS is required when new protocols or specifications that have particular performance requirements are designed before their performance properties have been demonstrated experimentally.

The need to validate performance arises, for example, in the case of media access protocols, where the medium constitutes a unique resource that has to be shared between the stations. The protocol used in such cases relies on algorithms that need to be evaluated (e.g. Ethernet, Token Ring and Aloha). The validation of such protocols should be based only on the configuration parameters defined in the relevant standards and should make no assumptions on the technology used.

The principle of validation is to determine whether certain criteria (for instance the response time) are matched under certain conditions (traffic levels, for example). To be able to perform such validation, the performance objectives should be precisely defined and recorded. An example of a document that specifies such figures is ETR 102 [7], "European digital cellular telecommunication system (Phase 2); Technical performance objectives", which is a part of the ETSI GSM technical specifications.

The validation of performance may require the development of a validation model that differs from the one used to validate the functional behaviour. It would clearly be interesting to use the same model for validating both function aspects and performance. Some research has been undertaken on this subject and results obtained by the Lund Institute of Technology are published in "Performance Simulation for SDT" [20].

There are different techniques and tools for the validation of performance. The SDT tool used within ETSI, for instance, incorporates a performance simulation tool that has been used to validate performance in radio networks.

6.3 Relationship between validation and the specification of conformance test suites

6.3.1 Validation of a testing specification

The conformance test specification methodology, ETS 300 406 [5], lists a number of quality criteria that a test specification has to meet before it can be standardised. The standard also defines what validation means in respect to a testing specification and the conditions for standardization of such testing specification. This chapter addresses only the relationship between the testing process and the validation process.

6.3.2 Validation during test suite development

When a standard reaches a sufficient level of maturity, the production of a test suite for this standard may be developed. This is a manual process carried out according to criteria defined in ISO/IEC IS 9646 [14]. ITU-T and ISO are also investigating methods for the automatic generation of test suites (see subclause 4.3).

Test suite development often identifies problems in an ETS. Manual test case derivation involves a thorough inspection of the specification (walkthrough) for which the test cases are intended. The elaboration of a test specification can contribute to the validation of a product standard. Ambiguities and inconsistencies can be easily identified if tests are precisely defined.

Automatic test case generation is a more advanced technique. A model similar to the one identified in subclause 6.1 is used for this. During test case generation, abnormalities in the specification may be identified. For example, if there is a deadlock in the specification, the test case generator may encounter this deadlock while trying to produce test cases for the transitions involved.

The time at which the testing activities begin is a compromise between two needs. Firstly, the production of a test suite generally represents a major effort and should not be launched too early. Secondly, as it helps to validate the standard, some overlap with the development activities is useful. Another reason to overlap these activities is that the product standard should be developed such that it is testable.

6.3.3 Reuse of validation activities in test suite development

The validation and testing activities have a number of common features. For example, similar test scenarios need to be produced for both. Taking advantage of these similarities, the effort of producing a test suite can be reduced. Output of the validation phase can be re-used to provide the basis for further test suite definition.

6.3.4 Combined approach

Subclauses 6.3.2 and 6.3.3 describe benefits that can be obtained by using the test suite to validate the model or conversely to use the validation results to build the test suite. The first approach is appropriate when the highest priority is to produce a test suite. The second approach implies production of a test suite only after the standard has been validated. This is a more solid base for test suite production implies a delay to the start of test production.

It is, however, more realistic to assume that there will be a certain amount of overlap in the validation and testing activities, with each contributing to the success of the other.

6.4 Planning for validation

In developing a draft ETS, the rapporteur will be greatly assisted by knowing not only what the specific technical requirements are which need to be met by the contents of the ETS but also what validation activity is to be applied to the finished draft. It follows that ETS validation has to be planned together with the development of the standard. It is necessary to define a comprehensive planning model for ETS validation and it would be desirable to have the following aspects considered:

- the type of validation that will be used;

  NOTE: The type of validation depends on the expected results of the standard development process. If the resultant ETS will specify function or behaviour using the ETSI standard development methodology, the object of validation will be an executable validation model. In that case formal validation supported with SDL tools has to be planned;

- the prerequisites for validation:
  - documentation;
  - procedures;

- the timing of validation activities:
  - in relation to ETS development;
  - in relation to Conformance Test Suite development;

- the equipment necessary for successful validation:
  - software for analysis and simulation;
  - processing platforms for validation tools;

- The validation team:
  - independent;
  - the development team;

- Documentation of validation activity:
  - validation results;
  - recommendations.

7 Requirements for a validation methodology

There is a need to document an ETSI methodology for validation that can be used as guidance for the validation task. The methodology for validation will be written as a component of the general methodology for technical preparation of standards (see annex A) and this requires further study.

7.1 Scope of validation

The practical application of the various validation techniques available and the different objectives of validation will be defined in detail in the methodology description.

The methodology will require the objectives of validating a particular ETS to be recorded before validation takes place.

7.2 Compatibility with the methodology for development

The methodology for validation will be written to complement the ETSI methodology for development of specifications, DTR/MTS-00013 [8].

The methodology will allow validation to overlap in time with checking that an ETS under development conforms to quality rules. The reason is that an interaction between development and validation is an effective use of effort and time.

A systematic validation methodology is facilitated by the use of "classified requirements" identified in the development methodology in a uniform way for all ETSs. The "classified requirements" are used as a basis for validation as shown in annex A.

7.3 Activities in the methodology

The methodology will be divided into one or more activities with clearly identifiable inputs and outputs.

There needs to be criteria for determining that an activity can start and criteria for determining when to terminate an activity. These criteria have 2 components:

- general criteria that apply in every case;
- criteria specific to a particular work item (or a class of similar work items).

Examples of general criteria are:

- the selection of options, such as can be found in an International Standardised Profile (ISP), to enable a validation model to be completed;
- the existence of a closed SDL validation model before dynamic analysis can commence.

The general criteria will be part of the methodology.

Examples of specific criteria are:

- the number of users to be involved in a multi-party call in the validation model;
- the percentage of inconclusive aspects that are acceptable when a final validation is carried out.

Specific criteria are specified for each work item project as part of the planning of the project. The methodology will contain advice on their selection.

The specific activities of validation will cover:

- informal validation (see subclause 5.1.3);

- creation of the validation model that will include:
    - determination of the content and scope of the validation model;
    - configuring the model to any PICS;
    - the definition of the limits in the model (see subclause 6.1.2);
    - modelling of unspecified functions;

- the creation or selection of scenarios ("test cases") for validation;

- analysis by use of the model, with scenarios for validation.

The methodology will describe the activity for scenario creation or selection. This description will include instructions, guidelines and rules for selection. The MSC in the ETS, the conformance test specification for the ETS, or new scenarios will be used (see subclause 6.3).

The methodology will have several different activities for validation (see subclause 5.1). It will also contain advice on the selection of analysis activities according to the following criteria:

- the subject of the ETS;
- the facilities and resources available;
- the criteria for the results of validation.

Each activity will be described in the methodology in terms of simple steps that are easy to carry out. An example of the description of the methodology will be included in DTR/MTS-00013 [8]. The description of steps in terms of instructions for operating a particular tool should be avoided.

7.4 Documentation of validation results

The methodology will define how to present both the validation activities performed and the results achieved. Wherever possible, the results will be presented in the form of lists or tables rather than prose.

When validation results in changes to the specification, an appropriate record of the defects found and the changes made should be maintained. In the final validation, all validation activities should be classified as valid, invalid or inconclusive and unresolved problems should be documented.

7.5 Skill basis

The skills required to carry out a work item project need to be identified. The methodology will give advice about the skills needed. Additional material that may help less experienced users to apply the methodology will be identified. This material will include style guides, tutorial courses and books on the languages and techniques used.

Annex A: Validation and SDL development

A.1 Relationship between validation methods and SDL methodology

The starting point in the development of an ETS is a Work Item that can vary from a simple expression of need, to references to substantial technical reports and existing standards. As well as the TA Working Procedures [1], which only give general guidelines, there needs to be more detailed technical direction available on the preparation of the technical content of standards. Such an enhancement to the existing methodology would be a significant benefit not only in simplifying the management of ETS development but also in assuring the quality of the resultant standard.

Within the technical preparation of a standard, different processes can be identified:

- development of the deliverable;
- production of the conformance test specifications corresponding to the deliverable (if it is a base standard);
- validation of the deliverable.

Figure A.1 shows the various rules and principles that are either published or planned. This ETR identifies the requirements for the Validation Techniques shown in the shaded box.

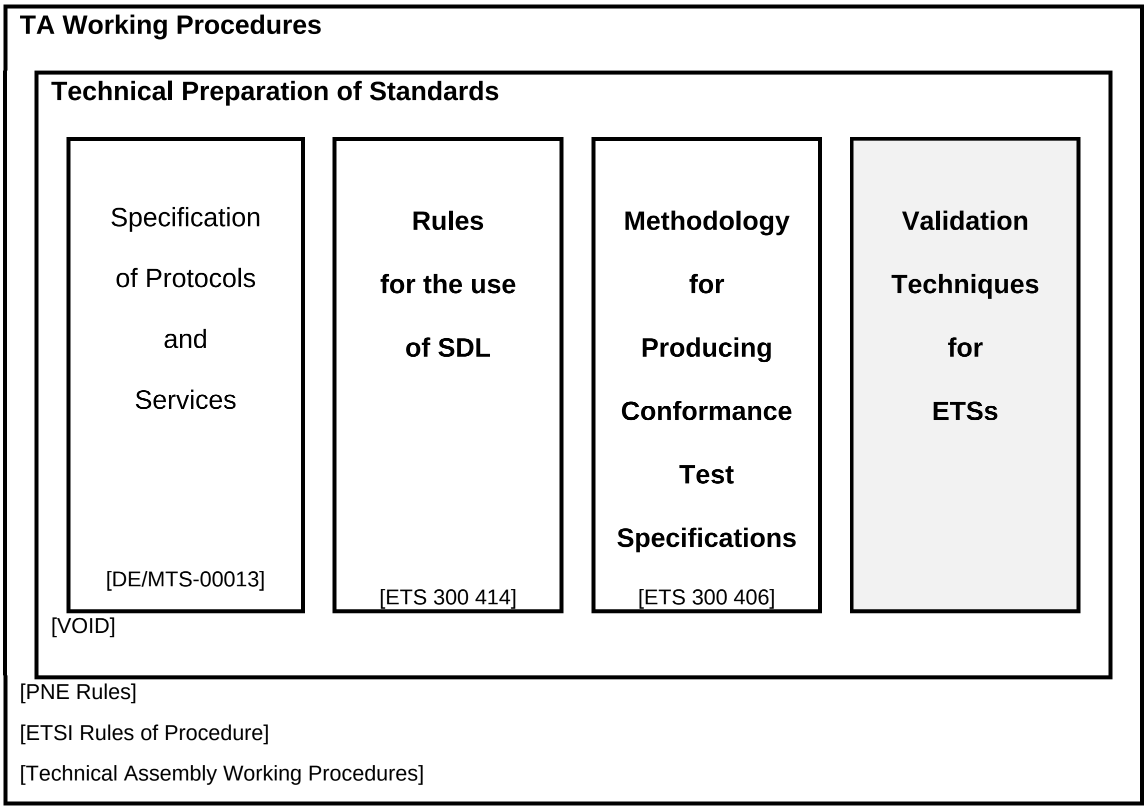
TA Working Procedures
Technical Preparation of Standards
Specification of Protocols and Services
[DE/MTS-00013]
Rules for the use of SDL
[ETS 300 414]
Methodology for Producing Conformance Test Specifications
[ETS 300 406]
Validation Techniques for ETSs
[VOID]
[PNE Rules]
[ETSI Rules of Procedure]
[Technical Assembly Working Procedures]

Figure A.1: Procedures and rules applying to technical preparation of ETSs

No specific documents or plans for the production of documents currently exist for the part of figure 1 marked as [VOID]. It represents procedures that are more technical or more detailed than the procedures in PNE Rules [3], ETSI Rules of Procedure [20] or Technical Assembly Working Procedures [1], but more general than DTR/MTS-00013 [8], ETS 300 414 [6], ETS 300 406 [5], and the validation methodology. It is expected these general procedures will include:

- the technical planning of work item projects;
- references to more detailed methodologies;
- producing, using and maintaining the contents of a work item library;
- the technical quality assurance of work item production.

The relationship between ETS development and validation is described in terms of processes where a process can be described as a sequence of actions that accepts inputs and produces outputs (see figure A.2). However, such processes require principles and procedures to control them, tools to support these principles and procedures, and the expertise and knowledge necessary to develop the raw inputs into the required output.

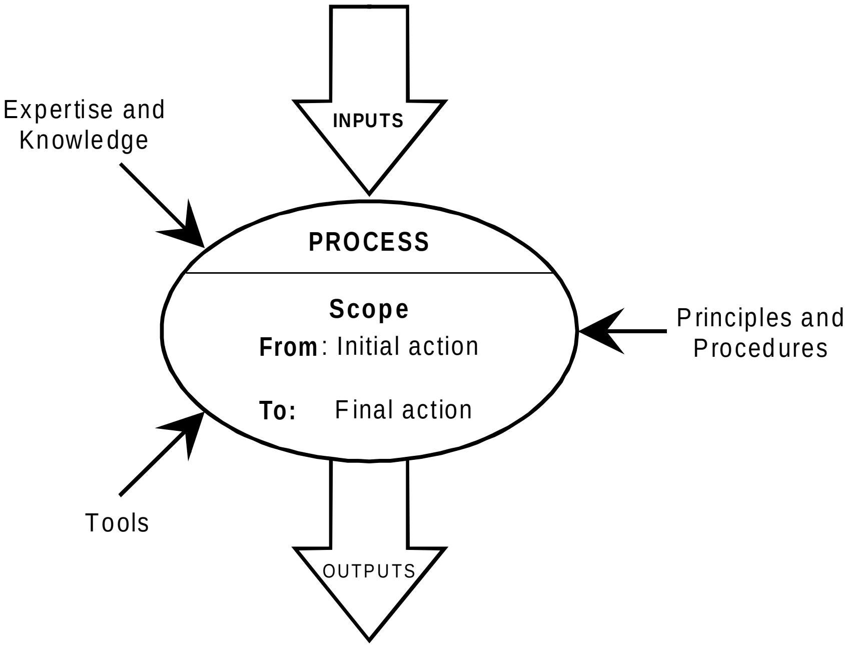

Figure A.2: A process

In almost all product development, the processes for marketing, design and management take place in parallel while also having dependencies on each other. The same is true in the production of ETSs. There are dependencies between the processes but, within the constraints imposed by these dependencies, processes can take place in parallel. The validation process, for example, can start as soon as there is something to validate in the draft ETS.

Figure A.3 shows the production of validated standards. The starting point is the process "Requirements Capture". The output of this process is a set of collected requirements that is the information referred to or implied by a Work Item. Requirements capture is not a well-defined process and covers the work carried out by a TC before creating a work item. It also covers the collection of further requirements and the clarification of existing requirements during the development of an ETS.

The collected requirements need to be classified, as will be described in DTR/MTS-00013 [8], so that the information is formally structured. Classified requirements form a sound basis for the development and subsequent validation of an ETS.

NOTE The collected requirements do not have to be completely classified before development starts, nor do the classified requirements have to be completed before validation starts. In both cases the processes can be started as soon as minimum criteria are met.

Validation of an ETS should be with respect to the classified requirements and to other ETSs and rules that are relevant to the ETS. These include:

- ETSs for functions that inter work with the functions specified in the ETS under validation;
- ETSs and other rules (such as PNE Rules [3] or certain TCR-TRs) to which the ETS under validation should conform;
- the corresponding conformance test specification for the ETS under validation.

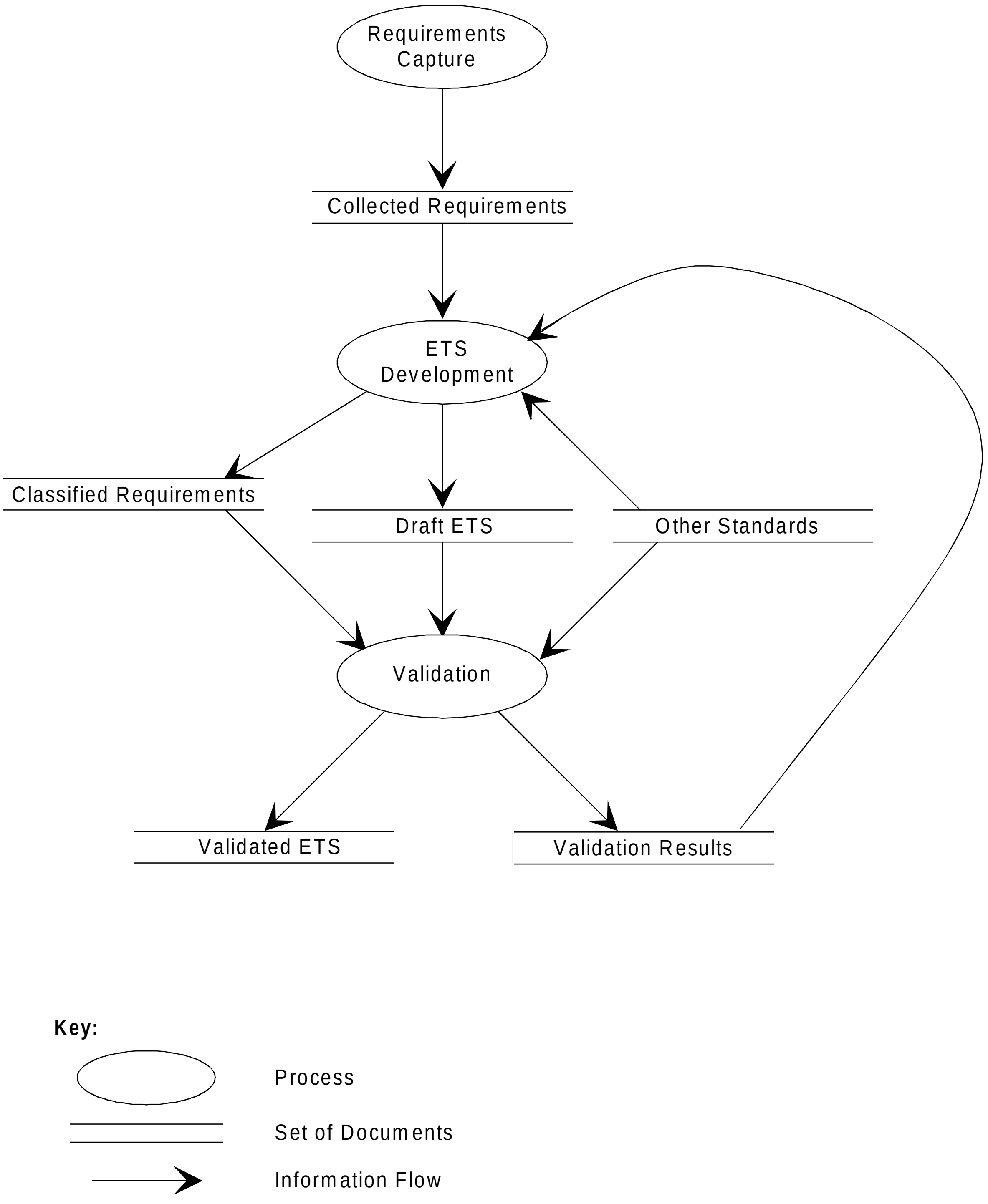

Figure A.3: The preparation of validated standards

Figure A.4 shows in generalised form, how the processes of ETS development and validation are associated in time. The shaded bars indicate work being done.

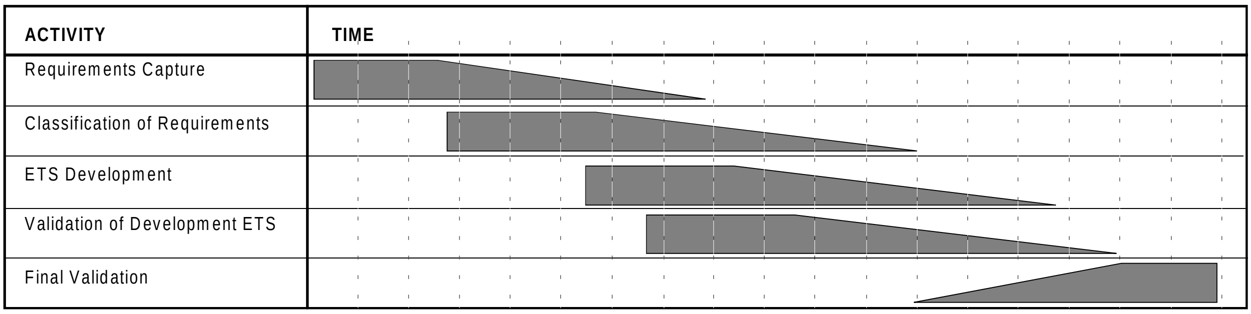

Figure A.4: Generalised time-line of the ETS development and validation processes

Validation of the contents of a draft ETS can begin as soon as some classified requirements and the corresponding sections of the ETS are available and this should continue throughout the development process. However, the final validation of an ETS cannot be completed until the ETS, itself, is complete. The purpose of this independent assessment is to establish that the ETS can be identified as satisfying all of the requirements expressed for it.

A.1.1 Classified requirements

The requirements for an ETS are most often expressed in unstructured prose either as notes on a Work Item Sheet or in a Technical Report. Sometimes the requirements are very brief but have wide implications. At other times, they are very expansive but can be reduced to a small number of simple statements.

Although not impossible, it would be very difficult to validate that a completed ETS conforms to its specific requirements if those requirements are not expressed in a unified and structured format.

The original informal requirements need to be reviewed and classified into a set of individual requirements which are:

- unambiguous;
- complete;
- correct;
- consistent:
  - within the context of the requirements of the ETS;
  - within the context of other published or draft ETSs.
- understandable;
- achievable;
- verifiable.

A.1.1.1 Examples of requirements

Table A.1 shows some examples of unstructured requirements that have been classified into requirements that can be validated in a completed ETS.

Table A.1: Examples of classified requirements derived from unstructured requirements

Original requirement	Classified requirement
Cordless Terminal Mobility should support all the supplementary services offered to wired terminal users . . . . Cordless Terminal Mobility should provide support for all private network basic services - 3,1kHz speech . . . . Cordless Terminal users shall be able interact with other users in a private network based on QSig protocols	Cordless Terminal Mobility is a set of services that . . . . do not specify any changes to the existing QSig basic service.
The Location Registration Service should support all radio access systems that are ETSI standards for Private Telecommunications Networks	The Location Registration is a service that . . . . supports the air interface protocols and identifier structures of the following standardised systems: – Digital European Cordless Telecommunication system, DECT, (ETS 300 175) – Cordless Telephony system, number 2, CT2, (ETS 300 131)

A.1.2 Draft ETS

A draft ETS is the output of the ETS production process. It is provided by the assigned rapporteur or project team and is the item to be validated within the validation process.

A.1.3 Other standards

All ETSs have to conform to the requirements specified in other standards. These requirements fall into two broad categories:

- those which specify the form of the ETS and the methodology to be followed during development;

  Examples:

  - PNE Rules [3];
  - ETS 300 387 - the method for specifying Private Telecommunication Network, PTN, supplementary services;
  - ITU-T Recommendation Z.100 [11] - SDL;
  - ETS 300 414 [6] - ETSI rules for testability and facilitation of validation;

- those which specify technical aspects with which the ETS has to be consistent.

  Examples:

  - An Integrated Services Digital Network (ISDN) supplementary service protocol specification needs to be consistent with the specification of the DSS1 basic call (ETS 300 102);

  - A stage 2 standard for a supplementary service specification needs to be consistent with the associated stage 1 standard and the stage 3 needs to be consistent with the stage 2.

These additional standards should all be considered as inputs to the validation process against which the subject ETS will be checked.

A.1.4 Validation results

At the end of the validation process, a set of results will be produced indicating the status of the subject ETS after validation. The validation results indicate:

- what validation has been performed;
- which aspects of the ETS were found to be "valid";
- which aspects of the ETS were found to be "not valid", and should include the basis for reaching this conclusion;
- which aspects of the ETS were found to be "inconclusive", and the basis for reaching this conclusion should be given.

If the validated ETS is not found to be "valid" then the responsible TC may be expected to provide additional requirements or clarification to ensure that the ETS is re-worked successfully.

A.1.5 Validated ETS

When all aspects of the draft ETS have successfully passed through the validation process, it is deemed to be validated and can be submitted to the responsible STC and TC for their approval.

A.2 Testability and validation rules

The rules for testability and facilitating validation, ETS 300 414 [6], are applicable to ETSs that use SDL, MSC or ASN.1 (when combined with SDL).

Checking that the rules have been correctly applied is an important preliminary step of the validation activity. Currently, this has to be performed by manual inspection, since none of the available tools checks the specific ETSI recommendations. The building of a useful validation model depends on the application of the rules as is the ability to build an applicable testing method.

Part of any validation training program should ensure that the standard developers are well aware of the existence and value of these rules. The application of the rules should also be monitored in order to verify their comprehensiveness. The experimentation within the framework of pilot projects will generate valuable information relating to whether the application of these rules is sufficient for validation.

Standards can be written in different formats, while still conforming to these rules. Future work should look at the definition of a common format.

Annex B: Machine readable standards

Currently ETSI standards are distributed in the form of hard copy documents. In this form, they can only be read and require considerable study by those who need to use them. It may be necessary for an implementer to master a set of standards in order to understand the intricacies of a complex system. Making standards available in machine readable form can significantly enhance their utility to the user. For example, searching a set of documents for all references to an entity becomes a trivial task.

The inclusion of SDL models in standards raises certain issues in this context. Components of a standard that are defined in SDL/GR can in principle be exercised. Machine readable standards that included executable models so that users of the standard could input the models into their own analysis tools would appear to have enormous potential value.

Microsoft Word has some limitations as a vehicle for producing and distributing documents in machine readable form. It can output text in a variety of formats, for input into other word processors, and as straight ASCII files. It has a significant limitation (shared with many other, similar tools) in the way it handles diagrams as opposed to text. Diagrams are included in Word documents as non-coded information. A simple test demonstrates that if the *Find* command is used to search a Word document for a particular phrase, it will only identify instances in the text of the document, and not those in the diagrams. Diagrams are maintained and filed in the form of bitmaps so that the text they contain is non-coded information.

This limits the use of Word documents. For example, suppose that it is desirable to produce a cross reference listing of identifiers used in a standard in order either to search it when one is used, or to improve the self consistency of the standard when it is being developed. Such a facility can only analyse the text and not the imbedded diagrams in a Word document. If the use of SDL diagrams in standards increases, an increasing fraction of the documents will consist of non-coded information that cannot be subject to any form of machine aided analysis or searching, even when the user has access to the document in machine readable form.

It would appear important to make the whole of a standards document accessible in a coded, machine readable form. This would probably imply including SDL/PR representations of the SDL diagrams as part of the coded document to overcome the limitations discussed above. This could make it possible to develop cross reference and other tools that could ease browsing and self consistency checks of both the text and SDL in a single standard as well as a whole set of standards. Validated standards that contained executable and accessible models of the defined functions would be of enormous value to implementers and have the potential of significantly reducing the time taken to transform them into useful implementations.

Focusing the production of standards on documents intended to be executed rather than merely read would contribute more to the quality of the standards than any other single change in the development process.

History

Document history	
April 1995	First Edition
February 1996	Converted into Adobe Acrobat Portable Document Format (PDF)

ISBN 2-7437-0109-9
Dépôt légal : Avril 1995