

Communication networks and systems in substations —

Part 6: Configuration description language for communication in electrical substations related to IEDs

The European Standard EN 61850-6:2004 has the status of a
British Standard

ICS 33.200

National foreword

This British Standard is the official English language version of EN 61850-6:2004. It is identical with IEC 61850-6:2004

The UK participation in its preparation was entrusted to Technical Committee PEL/57, Power system control and associated communications, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 21 June 2004

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 144, an inside back cover and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

© BSI 21 June 2004

ISBN 0 580 43928 3

Communication networks and systems in substations
Part 6: Configuration description language for communication
in electrical substations related to IEDs
(IEC 61850-6:2004)

Réseaux et systèmes de communication
dans les postes
Partie 6: Langage pour la description
de configuration pour la communication
dans les postes électriques,
entre les dispositifs électroniques
intelligents (IED)
(CEI 61850-6:2004)

Kommunikationsnetze und -systeme
in Stationen
Teil 6: Sprache für die Beschreibung
der Konfiguration für die Kommunikation
in Stationen mit intelligenten
elektronischen Geräten (IED)
(IEC 61850-6:2004)

This European Standard was approved by CENELEC on 2004-05-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

The text of document 57/693/FDIS, future edition 1 of IEC 61850-6, prepared by IEC TC 57, Power systems management and associated information exchange, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61850-6 on 2004-05-01.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2005-02-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2007-05-01

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 61850-6:2004 was approved by CENELEC as a European Standard without any modification.

CONTENTS

1	Scope	6
2	Normative references.....	6
3	Terms and definitions	7
4	Abbreviations.....	7
5	Intended engineering process with SCL.....	8
6	The SCL object model	10
6.1	General	10
6.2	The substation model	13
6.3	The product (IED) model.....	14
6.4	The communication system model	15
6.5	Modelling redundancy	16
7	SCL description file types.....	16
8	The SCL language	17
8.1	Specification method	17
8.2	SCL language extensions	19
8.3	General structure.....	22
8.4	Object and signal designation	23
9	The SCL syntax elements	26
9.1	Header	26
9.2	Substation description	29
9.3	IED description	41
9.4	Communication system description	65
9.5	Data type templates.....	71
	Annex A (normative) SCL syntax: XML schema definition	83
A.1	Base types.....	83
A.2	Substation syntax	94
A.3	Data type templates	99
A.4	IED capabilities and structure.....	102
A.5	Communication subnetworks	110
A.6	Main SCL.....	115
	Annex B (normative) SCL enumerations according to IEC 61850-7-3 and IEC 61850-7-4	116
	Annex C (informative) Syntax extension examples.....	122
C.1	Extension syntax for drawing layout coordinates.....	122
C.2	Extension syntax for maintenance	124
	Annex D (informative) Example	125
D.1	Example specification	125
D.2	Example SCL file contents	127
	Annex E (informative) XML schema definition of SCL variants	136
	Annex ZA (normative) Normative references to international publications with their corresponding European publications	142



Figure 1 – Reference model for information flow in the configuration process.....	9
Figure 2 – SCL object model	11
Figure 3 – Configuration example	13
Figure 4 – UML diagram overview of SCL schema	19
Figure 5 – Elements of the signal identification as defined in IEC 61850-7-2	24
Figure 6 – Elements of the signal name using functional naming.....	25
Figure 7 – Elements of the signal name using product naming	25
Figure 8 – Names within different structures of the object model.....	26
Figure 9 – UML diagram of Header section.....	27
Figure 10 – UML diagram of Substation section	29
Figure 11 – UML diagram for equipment type inheritance and relations	34
Figure 12 – IED structure and access points.....	42
Figure 13 – UML description of IED related schema part – base	43
Figure 14 – UML description of IED related schema part for Control blocks.....	44
Figure 15 – UML description of IED related schema part – LN definition.....	45
Figure 16 – UML diagram overview of the Communication section	66
Figure 17 – UML overview of DataTypeTemplate section	72
Figure C.1 – Coordinate example	122
Figure D.1 – T1-1 Substation configuration.....	125
Figure D.2 – T1-1 Communication configuration	126
Figure D.3 – T1-1 Transformer bay.....	127
Table 1 – The files composing the XML schema definition for SCL.....	18
Table 2 – Attributes of the Private element	21
Table 3 – Attributes of the Header element.....	27
Table 4 – Attributes of the History item (Hitem) element	28
Table 5 – Primary apparatus device type codes	36
Table 6 – Attributes of the Terminal element.....	37
Table 7 – Attributes of the SubEquipment element.....	38
Table 8 – Attributes of the LNode element	38
Table 9 – Attributes of the IED element	46
Table 10 – List of service capabilities and setting elements and attributes	47
Table 11 – Attributes of the Access point element.....	49
Table 12 – Attributes of the IED server element.....	50
Table 13 – Attributes of the Authentication element	51
Table 14 – Attributes of the LDevice element.....	51
Table 15 – Attributes of the LN0 element.....	52
Table 16 – Attributes of the LN element.....	53
Table 17 – Attributes of the DOI element	54
Table 18 – Attributes of the DAI element	54
Table 19 – Attributes of the SDI element	55
Table 20 – Attributes of the DataSet element.....	55
Table 21 – Attributes of the FCDA element.....	56

Table 22 – Attributes of the report control block element.....	57
Table 23 – Attributes of the RptEnabled element	58
Table 24 – Attributes of the ClientLN element.....	59
Table 25 – Attributes of the log control block element	60
Table 26 – Attributes of the GSE control block element.....	61
Table 27 – Attributes of the sampled value control block element.....	62
Table 28 – Attributes of the Smv Options element	62
Table 29 – Attributes of the setting control block element	63
Table 30 – Attributes of the Input/ExtRef element	64
Table 31 – Attributes of the Association element	65
Table 32 – Attributes of the Subnetwork element	67
Table 33 – Attributes of the ConnectedAP element	68
Table 34 – Attributes of the GSE element	69
Table 35 – Attributes of the SMV element.....	70
Table 36 – PhysConn P-Type definitions	70
Table 37 – Template definition elements	74
Table 38 – Attributes of the LNodeType element.....	74
Table 39 – Attributes of the DO element	75
Table 40 – Attributes of the DOType element.....	75
Table 41 – Attributes of the SDO element.....	75
Table 42 – Data type mapping	76
Table 43 – Attribute value kind (Valkind) meaning	77
Table 44 – Attributes of the DA element	78
Table 45 – Attributes of the BDA element	80
Table 46 – Attributes of the EnumType element.....	81

COMMUNICATION NETWORKS AND SYSTEMS IN SUBSTATIONS –

Part 6: Configuration description language for communication in electrical substations related to IEDs

1 Scope

This part of the IEC 61850 series specifies a file format for describing communication related IED (Intelligent Electronic Device) configurations and IED parameters, communication system configurations, switchyard (function) structures, and the relations between them. The main purpose of this format is to exchange IED capability descriptions, and SA system descriptions between IED engineering tools and the system engineering tool(s) of different manufacturers in a compatible way.

The defined language is called Substation Configuration description Language (SCL). The IED and communication system model in SCL is according to IEC 61850-5 and IEC 61850-7-x. SCSM specific extensions or usage rules may be required in the appropriate parts.

The configuration language is based on the Extensible Markup Language (XML) version 1.0.

This standard does not specify individual implementations or products using the language, nor does it constrain the implementation of entities and interfaces within a computer system. This part of the standard does not specify the download format of configuration data to an IED, although it could be used for part of the configuration data.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61346-1:1996, *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules*

IEC 61346-2:2000, *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 2: Classification of objects and codes for classes*

IEC 61850-2, *Communication networks and systems in substations – Part 2: Glossary*

IEC 61850-5, *Communication networks and systems in substations – Part 5: Communication requirements for functions and device models*

IEC 61850-7-1, *Communication networks and systems in substations – Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models*

IEC 61850-7-2, *Communication networks and systems in substations – Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)*

IEC 61850-7-3, *Communication networks and systems in substations – Part 7-3: Basic communication structure for substation and feeder equipment – Common data classes*

IEC 61850-7-4, *Communication networks and systems in substations – Part 7-4: Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes*

IEC 61850-8-1, *Communication networks and systems in substations – Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*

IEC 61850-9-1, *Communication networks and systems in substations – Part 9-1: Specific Communication Service Mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link*

IEC 61850-9-2, *Communication networks and systems in substations – Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3*

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

Extensible Markup Language (XML) 1.0, W3C, available at <http://www.w3.org/TR/2000/REC-xml-20001006>

Namespaces in XML, W3C, available at <http://www.w3.org/TR/1999/REC-xml-names-19990114>

XML Schema Part 0: Primer, W3C, available at <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502>

XML Schema Part 1: Structures, W3C, available at <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>

XML Schema Part 2: Datatypes, W3C, available at <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>

RFC 1952, *GZIP file format specification version 4.3*, RFC, available at <http://www.ietf.org/rfc/rfc1952.txt>

RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, RFC, available at <http://www.ietf.org/rfc/rfc2045.txt>

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61850-2 apply.

4 Abbreviations

In general, the glossary and abbreviations defined in IEC 61850-2 apply. The following abbreviations are either special for this part of the standard, or particularly useful for understanding this part and are repeated here for convenience.

BDA	Basic Data Attribute, that is not structured
CIM	Common Information Model for energy management applications
DAI	Instantiated Data Attribute
DO	DATA in IEC 61850-7-2, data object type or instance, depending on the context
DOI	Instantiated Data Object (DATA)
DTD	Document Type Definition for an XML document
ID	Identifier
IED	Intelligent Electronic Device
LDInst	Instantiated Logical Device

LNInst	Instantiated Logical Node
MSV	Multicast Sampled Value
MsvID	ID for MSV (Multicast Sampled Value)
RCB	Report Control Block
SCL	Substation Configuration description Language
SDI	Instantiated Sub DATA; middle name part of a structured DATA name
UML	Unified Modelling Language according to http://www.omg.org/uml
URI	Universal Resource Identifier
UsvID	ID for USV (Unicast Sampled Value)
XML	Extensible Markup Language

5 Intended engineering process with SCL

Engineering of a substation automation system may start either with the allocation of functionally pre-configured devices to switchyard parts, products or functions, or with the design of the process functionality, where functions are allocated to physical devices later, based on functional capabilities of devices and their configuration capabilities. Often a mixed approach is preferred: a typical process part such as a line bay is pre-engineered, and then the result is used within the process functionality as often as needed. For SCL, this means that it must be capable of describing:

- A system specification in terms of the single line diagram, and allocation of logical nodes (LN) to parts and equipment of the single line to indicate the needed functionality.
- Pre-configured IEDs with a fixed number of logical nodes (LNs), but with no binding to a specific process – may only be related to a very general process function part.
- Pre-configured IEDs with a pre-configured semantic for a process part of a certain structure, for example a double busbar GIS line feeder.
- Complete process configuration with all IEDs bound to individual process functions and primary equipment, enhanced by the access point connections and possible access paths in subnetworks for all possible clients.
- As item d) above, but additionally with all predefined associations and client server connections between logical nodes on data level. This is needed if an IED is not capable of dynamically building associations or reporting connections (either on the client or on the server side).

Case e) is the complete case. Both cases d) and e) are the result after SAS engineering, while case a) is a functional specification input to SAS engineering, and b) and c) are possible results after IED pre-engineering.

The scope of SCL as defined in this standard is clearly restricted to these purposes:

- 1) SAS functional specification (point a) above),
- 2) IED capability description (points b) and c) above), and
- 3) SA system description (points d) and e) above)

for the purpose of system design, communication engineering and the description of the engineered system communication for the device engineering tools in a standardised way.

This is reached by defining an object model describing the IEDs, their communication connections, and their allocation to the switchyard, and a standardized way to describe how this model shall be represented in a file to be exchanged between engineering tools. The resulting object model could also be the base for other engineering tasks, possibly with some additions. Therefore, and because of the additional needs of SCSMs, this standard considers the language as defined here as the core model, and defines how extensions of this core model for SCSMs as well as other (engineering) purposes can be done in a standardised way.

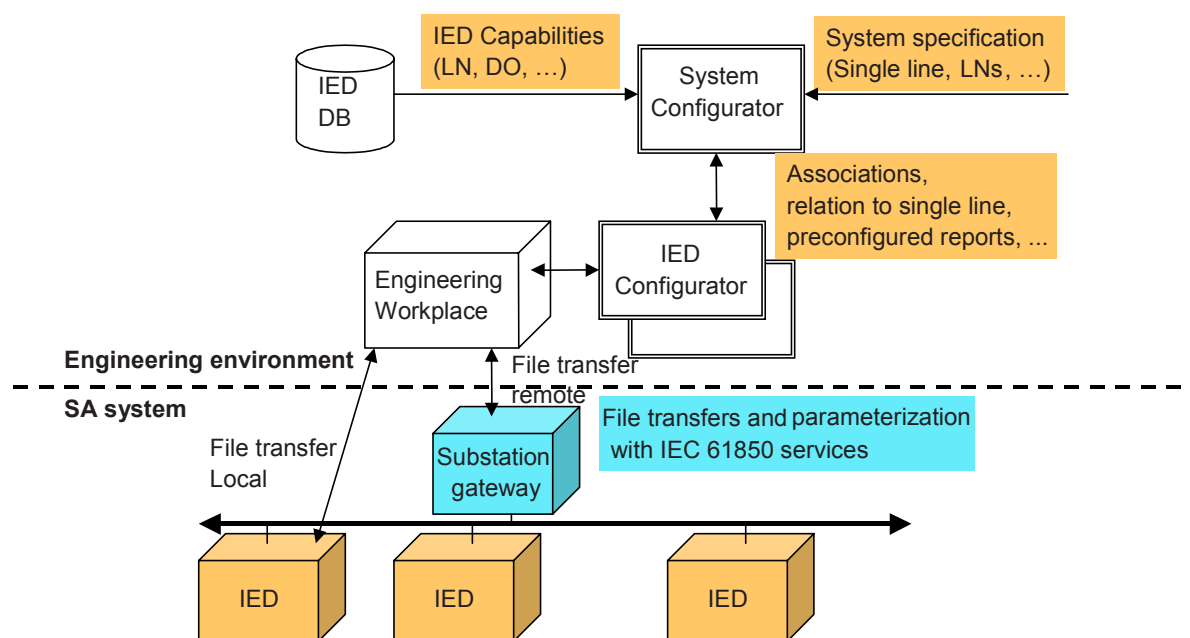
Figure 1 explains the usage of SCL data exchange in the above-mentioned engineering process. The shaded text boxes above the dashed line indicate where SCL files are used. The text box *IED capabilities* corresponds to a result of steps b) and c) above, the text box *System specification* corresponds to step a) above, the text box *Associations...* at the right to steps d) or e) above.

The IED Configurator is a manufacturer-specific tool that shall be able to import or export the files defined by this part of IEC 61850. It provides IED-specific settings and generates IED-specific configuration files, or it loads the IED configuration into the IED.

An IED shall only be considered compatible in the sense of the IEC 61850 series, if:

- It is accompanied either by an SCL file describing its capabilities, or by a tool, which can generate this file from the IED.
- It can directly use a system SCL file to set its communication configuration, as far as setting is possible in this IED (i.e. as a minimum, its needed addresses), or it is accompanied by a tool which can import a system SCL file to set these parameters to the IED.

The System Configurator is an IED independent system level tool that shall be able to import or export configuration files defined by this part of IEC 61850. It shall be able to import configuration files from several IEDs, as needed for system level engineering, and used by the configuration engineer to add system information shared by different IEDs. Then the system configurator shall generate a substation related configuration file as defined by this part of IEC 61850, which may be fed back to the IED Configurator for system related IED configuration. The System Configurator should also be able to read a System specification file for example as a base for starting system engineering, or to compare it with an engineered system for the same substation.



IEC 195/04

Figure 1 – Reference model for information flow in the configuration process

The part of Figure 1 below the dashed line indicates the ways in which IED configuration data produced by means of the IED configurator can be brought into the IED. This can be done by:

- local file transfer from an engineering workstation connected locally to the IED. This file transfer is beyond the scope of this standard.
- remote file transfer for example by the file transfer method of IEC 61850-7-2. The file format is not defined within this standard, but naturally SCL format is a possible choice.
- access services to parameter and configuration data defined according to IEC 61850-7-2. In this case, the standardised methods according to IEC 61850-7-x shall be used.

NOTE It is not in the scope of this standard to define any details of concrete software tools, which support an engineer in doing the intended engineering process with SCL described above. Both the system configurator as well as the IED configurator introduced above are also conceptual tools to illustrate the use of different SCL file variants in the engineering process. Each manufacturer is completely free to find the best way in supporting engineers by a specific software tool. Also completely free is the way, in which software tools for the above described engineering process with SCL will store manufacturer specific internal parameters for IEDs and SA system aspects, which are not in the scope of IEC 61850 (e.g. the relation of logical data to pins on a physical board), and how they relate them to the IEC 61850 data model.

6 The SCL object model

6.1 General

The SCL in its full scope describes a model of

- The primary (power) system structure: which primary apparatus functions are used, and how the apparatus are connected. This results in a designation of all covered switchgear as substation automation functions, structured according to IEC 61346-1.
- The communication system: how IEDs are connected to subnetworks and networks, and at which of their communication access points (communication ports).
- The application level communication: how data is grouped into data sets for sending, how IEDs trigger the sending and which service they choose, which input data from other IEDs is needed.
- Each IED: the logical devices configured on the IED, the logical nodes with class and type belonging to each logical device, the reports and their data contents, the (pre-configured) associations available; and which data shall be logged.
- Instantiable logical node (LN) type definitions. The logical nodes as defined in IEC 61850-7-x have mandatory, optional and user defined DATA (here abbreviated DO) as well as optional services, and are therefore not instantiable. In this document, instantiable LNTypes and DOTypes are defined as templates, which contain the really implemented DOs and services.
- The relations between instantiated logical nodes and their hosting IEDs on one side and the switchyard (function) parts on the other side.

SCL allows the specification of user defined DOs as an extension of standard LN classes as well as completely user-defined LNs according to the rules of IEC 61850-7-4. This means that the appropriate name space attributes shall be defined in the logical node types, and their value shall appear in the SCL file.

A SCL file describes an instance of the model in a serialized form and standardized syntax. However its semantic can only be fully understood by reference to the model itself, i.e. it is independent from the syntax. This Clause therefore gives an overview of the model by using UML notation. The next Clauses then define how an instance of the model is formally described in SCL.

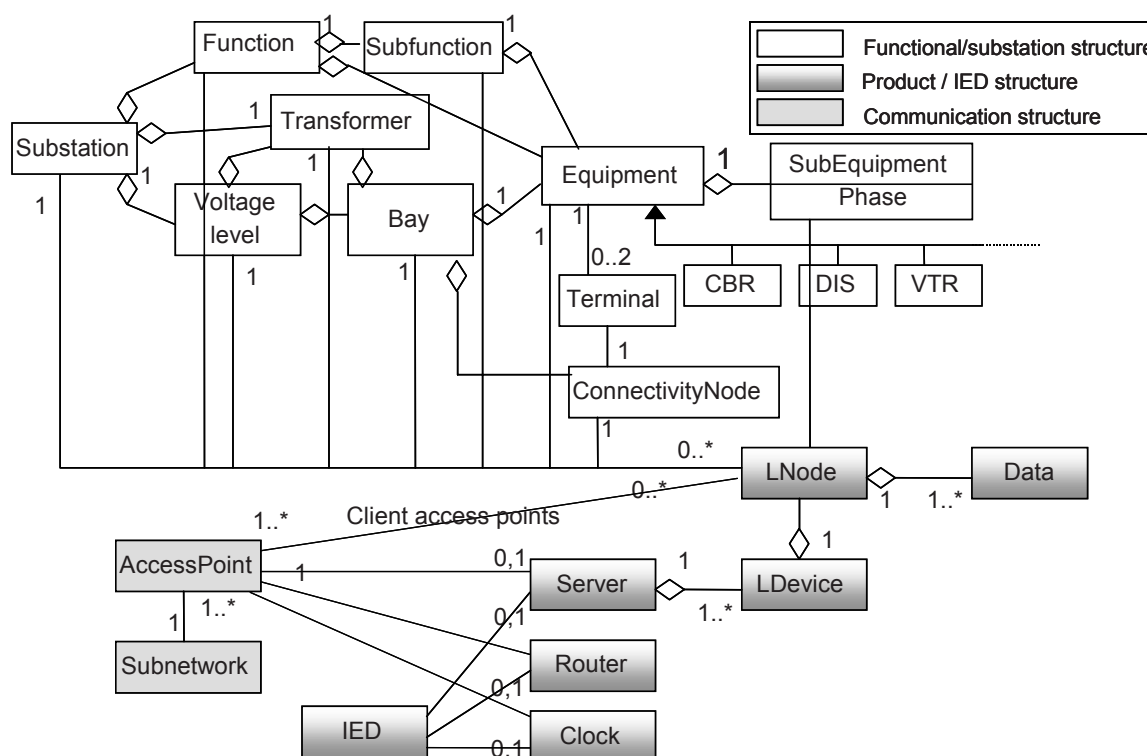
The UML object model is contained in Figure 2. Note that it is not complete in the modelling sense, i.e. it does not show any superclasses from which the used classes may be derived, no attributes etc. It restricts itself to those concrete object types that are used within a SCL instance file, in case of the substation related part, mainly for the purpose of functional designation. Furthermore it does not contain the levels below DATA (DOs), which are structurally defined in IEC 61850-7-2 and whose SCL description is defined in the DataTypeTemplates Clause.

The object model has three basic parts:

- 1) Substation: this part describes the switchyard equipment (process devices) in the functional view according to IEC 61346-1, their connection on single line level (topology), and the designation of equipment and functions.
- 2) Product: this stands for all SA product-related objects such as IEDs and logical node implementations.
- 3) Communication: this contains communication related object types such as subnetworks and communication access points, and describes the communication connections between IEDs as a base for communication paths between logical nodes as clients and servers.

Additionally, the data type template section allows, in a type-oriented (i.e. reusable) way, the specification of which data and attributes really exist in an IED. A logical node type as specified there is an instantiable template of the data of a logical node.

More model details contained in SCL, for example the structure within the logical nodes, are described in IEC 61850-7-x.



IEC 196/04

Figure 2 – SCL object model

The substation part and the product part in itself form hierarchies, which are used for naming and can be mapped to the functional and product structures according to IEC 61346 (all parts). The communication model part just contains the communication connection relations of IEDs to subnetworks, between subnetworks by means of routers at an IED, and the placement of master clocks at the subnetworks for time synchronisation. The modelling of gateways is not especially considered here. A gateway which is an IEC 61850 server has to be modelled like any other IEC 61850 compliant IED. The Proxy DO in the LPHD logical node allows to specify if a hosted LD is an image of another IED, or belongs to the hosting IED. A gateway being an IEC 61850 compliant client should host an ITCI logical node.

As can easily be seen from Figure 2, the logical node (abbreviated as LN or LNode) is the transition object, which is used to connect the different structures. This means that the LN instance as a product also has a functional aspect within the switchyard functionality and a communication aspect as a client or as a server within the substation automation system.

The substation functional objects as well as the product related objects are hierarchically structured. Each higher level object consists of lower level objects. This hierarchy is reflected in the designation structure of the objects according to IEC 61346-1. The function structure of IEC 61346-1 shall be used, and the designation coding of IEC 61346-2 should be used in the substation objects, while the IEC 61346-1 product structure should be used for IED designation structure and the IEC 61346-2 codes for the name values.

In SCL, it is foreseen that within each structure for nearly all objects, two kinds of designation are possible:

- A name is used as (a hierarchical part of) a technical key to designate the object. Each object within a hierarchy has an attribute *name*, which contains its identification within this level of the hierarchy. Technical keys are used in technical documentation for building and maintaining the system, or for automatic processing of engineering related information. This designation is also used in SCL to describe links between different model objects. In this case, as far as possible, the attribute containing the link gets a name of the form *<Targettype>Name*, for example *daName* for a link to a DATA attribute. This *name* relates to and is mostly identical to what is called *name* in IEC 61850-7-2.
- A description part is used as (a hierarchical part of) an operator- or user-related object identification. An object within a hierarchy has an attribute *desc*, which contains its textual description part within the hierarchy. Textual identifications are for example used in operator interfaces and operator manuals.

NOTE The desc SCL attribute is used at the engineering time, and identifies a (functional) object at its hierarchy level. The IEC 61850 d DATA attribute is used for describing data, and could also be read online. The contents of desc attributes could be used to generate a project specific (SCD) d text from a template (ICD) d text. This is however, not standardized.

A reference within SCL is, as defined in IEC 61850-7-2, a unique identification of an object, containing as a path the concatenation of all names in the hierarchy levels above, up to the level of the object. For the connection of power system equipment within a single line diagram, this path is used explicitly, while for other references it is used implicitly by stating only missing name parts. For forming names according to IEC 61850-7-2, the term *instance* with the abbreviation *inst* is also used. It is a part of a IEC 61850-7-2 name, making the full name unique within this level (see examples in 8.4).

The following Clauses describe the different parts of the model, their meaning and respective usage. Object attributes are mentioned here only if necessary for the understanding of the model. Further object attributes are described later in the SCL definition. Further model details belonging to IEC 61850-7-x and especially explained in IEC 61850-7-1 and IEC 61850-7-2 are purposely not shown here. The name model of the switchyard functionality is however only found in this part of IEC 61850, and therefore shown as far as used within this part of IEC 61850.

Figure 3 shows an instance of this model: a simple example of a SA system used for a switchyard. The naming is performed according to the IEC 61346 series. The switchyard has a 110 kV voltage level E1. It is a double bus bar system with two line bays =E1Q1 and =E1Q3, and a bus coupler =E1Q2. The IEDs are already assigned to switchyard functionality (for example the bay controller -E1Q1SB1 as a product is assigned to bay =E1Q1, and its LN CSWI1 controls the circuit breaker =E1Q1QA1 via the LN XCBR1 on the IED -E1Q1QA1B1). Observe that in IEC 61346-1 terms, here the bay is a transition object, i.e. it has a function (= sign, at switchyard level), and it is considered to be as product a part of the switchyard. This transition is seen in an SCL description only in the name structure of the IED name. Only the transition at the logical node is modelled explicitly. Figure 3 shows with the – (Minus) sign the product-related designation. The functional name is not repeated. The station level communication subnetwork is named W1. There are three additional subnetworks at process level (W2, W3, W4). Access points are seen in the picture, but their designations are not shown. Logical devices and servers are also not shown in the picture. This means especially that dynamic connections such as associations are not shown.

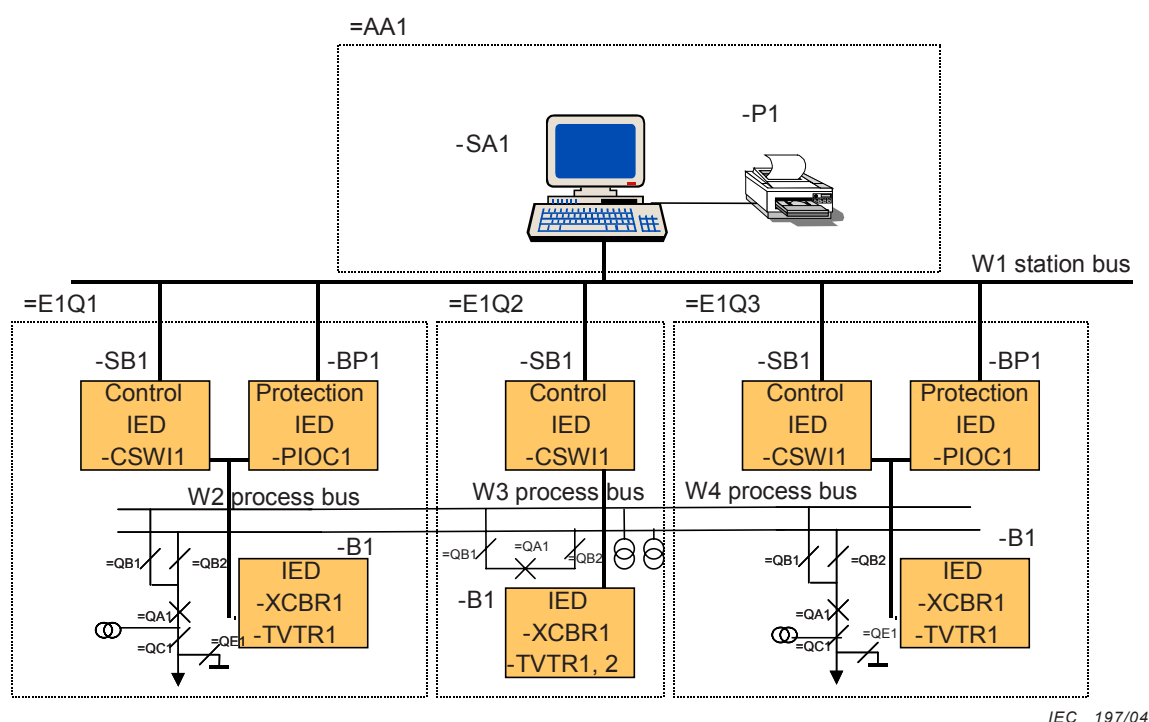


Figure 3 – Configuration example

6.2 The substation model

The substation model (upper part of Figure 2) is an object hierarchy based on the functional structure of the substation. Although each object is self-contained, its reference designation is derived from its place in the hierarchy. Because LNs perform functions within the complete context of the substation, they can be attached as functional objects at each substation function level. Typically, a switch controller LN is attached to a switching device, while a measuring LN is attached to the bay, which delivers the measurands, and transformer related LNs are attached to the appropriate transformer.

NOTE 1 In the CIM model measurands are allocated to primary device terminals. This is however a topological allocation, while the allocation in SCL in first line serves functional naming. However, if the single line topology is modelled completely, by means of the transformers (VTR, CTR) and their data acquisition nodes (TVTR, TCTR) also some primary device terminal in the topology can be found to which the measurands belong according to the CIM model.

The purpose of the substation model is

- to relate a logical node and its function to a substation function (substation part or equipment or subequipment);

- to derive a functional designation for the logical node from the substation structure.

The following substation objects of the functional structure (in hierarchical order) are used in the SCL model, analogue to the CIM model for energy management systems. More background information on these terms can be found in IEC 61850-2:

Substation	the object identifying a whole substation.
VoltageLevel	an identifiable, electrically connected substation part having an identical voltage level.
Bay	an identifiable part or subfunction of the switchyard (substation) within one voltage level.
Equipment	an apparatus within the switchyard, for example circuit breaker, disconnector, voltage transformer, power transformer winding etc. The single line diagram of a switchyard shows the electrical connections between these primary devices. Connectivity node objects model these connections. Therefore, each primary device can contain at its terminals references to the connectivity nodes to which it is connected. At single line level, one or two terminals (connections) are normally sufficient.
SubEquipment	a part of an Equipment, which might especially be one phase of a three-phase equipment.
ConnectivityNode	the (electrical) connectivity node object connecting different primary devices. Typical connectivity node examples are: connecting nodes within a bay, bus bars connecting several bays in the same voltage level, lines connecting bays in different substations. See also Equipment above.
Terminal	an electrical connection point of a primary apparatus at single line level. A terminal can be connected to a ConnectivityNode. Within SCL terminals can be explicitly named, or exist implicitly.

The *PowerTransformer* is special equipment, which can hierarchically be located below Substation, VoltageLevel or Bay. It contains Transformer windings as equipment, which might again have a relation to a tap changer.

NOTE 2 Observe that the hierarchical structure is mainly used for functional designations. If substructures of bays are needed, this can be introduced by appropriate bay names. If, for example, a bay B1 is structured into sub-bays SB1 and SB2, this would in the SCL model lead to two bays named B1.SB1 and B1.SB2. If logical nodes are also attached to the B1 structure level, then B1 can be introduced as a third bay.

The *Function* and *SubFunction* levels shown in Figure 2 can be used to model designations for any needed process part, which does not belong to the power system, such as building supervision and fire fighting systems.

6.3 The product (IED) model

Products consisting of hardware or software implement the functions of the switchyard. The scope of SCL from the product side only covers the hardware devices (called IEDs) that form the substation automation system, and therefore restrict the model to them. Primary devices as products are outside the scope of SCL, only their functional side is modelled by the substation structure for functional naming purposes.

IED	a substation automation device performing SA functions by means of logical nodes (LNs). It normally communicates via a communication system with other IEDs in the SA system.
Server	a communication entity within an IED according to IEC 61850-7-x. It allows access via the communication system and its only access point to the data of the logical devices and logical nodes contained in the server.
LDevice	a logical device (LD) according to IEC 61850-7-2 that is contained in a server of an IED.
LNode	a logical node (LN) implementation according to IEC 61850-5 and IEC 61850-7-2, contained in a logical device of an IED. The LN contains Data

(DO), which other logical nodes request, and it may need DOs contained in other LNs to perform its function. The *offered DOs* (server capability) are described in SCL. The *needed DOs* (LN client side) are determined by the function (LN) implementation and therefore configured by the IED configuration tool respectively by the engineer, which plans the system. SCL also allows their description, so that a data flow on data level between LNs can be modelled.

DO the DATA contained in the LNs according to IEC 61850-7-x.

NOTE Figure 2 shows with its LNode class the LN object, whose instances can be referenced or represented in SCL in two ways. The *LNode* element resides in the Substation structure, while the *LN* element resides in the IED structure.

This part of the standard additionally introduces as additional IED functions:

- a *Router* function on an IED. This is a function of the communication network, therefore it is described in 6.4.
- a *Clock* function to indicate where a subnetwork master clock is located.

6.4 The communication system model

The communication model is, in contrast to the others, not a hierarchical model. It models the logically possible connections between IEDs at and across subnetworks by means of access points. A subnetwork is seen at this description level only as a connecting node between access points, not as a physical structure. A logical device or a client of an IED is connected to a subnetwork by means of an access point, which may be a physical port or a logical address (server) of the IED. Client LNs use the address attribute of the access point to build up associations to servers on other IEDs respective to the LNs contained on the logical devices of these IEDs.

Although subnetworks only model logically possible connections, a correlation to the physical structure can be built up by appropriate naming of subnetworks and access points, and by the relation of access points to (one or more) physical connection points. The access points are the matching elements (transition objects) of both this communication model and the physical implementation of the communication system. The description and maintenance of the physical structure is beyond the scope of the core SCL.

Subnetwork	a connecting node for direct (link layer) communication between access points. It might contain telegram filtering on bridge level, but no routing on network level. All access points connected to a subnetwork can communicate with all others on the same subnetwork with the same protocol. SCSMs may define restrictions to this, for example if the stack implements a master-slave bus. The subnetwork as used here is a logical concept. Several logical subnetworks with different higher layer protocols could for example be used on the same physical bus to allow mixing of higher-level protocols on the same physical (lower) layer(s).
Access point	a communication access point of the logical device(s) of an IED to a subnetwork. There is at most one connection between a logical device and a subnetwork on this logical modelling level. An access point may, however, serve several logical devices, and the logical nodes contained in a logical device may, as clients, use several access points to connect to different subnetworks. Typically, a switch controller LN may get data as a client from a process bus (IEC 61850-9-x), and provide data as a server to the inter-bay bus (IEC 61850-8-1). In the terminology of IEC 61850-7-x, an access point may be used by a server, by a client, or by both. Furthermore, the same (logical) access point might support different physical access ports, for example an Ethernet connection and a serial PPP based connection to the same higher level (TCP/IP) access point and to the same server.
Router	Normally, clients connected to a subnetwork only have access to servers connected to that subnetwork. The router function extends access to servers connected to another subnetwork at another access point of that subnetwork.

hosts the router function. However, a router restricts the access to those services which use a networking layer, all other services such as GSE, sampled values and time synchronisation messages are not allowed to cross it.

Clock a master clock at this subnetwork, which is used to synchronize the internal clocks of all (other) IEDs connected to this subnetwork.

Routers and clocks are connected to a Subnetwork via their access points.

6.5 Modelling redundancy

Redundancy can be introduced to enhance the safety or availability of a system, and at different levels of the system:

- **IED internal:** this is beyond the scope of the IEC 61850 series, and therefore not describable with SCL. It is hidden in the IED HW/SW and externally visible just by error messages if something has failed. IED specific DATA might have to be introduced for these error indications.
- **Communication system level:** this is below the level described in the core SCL. Even if the communication system is doubled, but below the addressing level provided for a logical access point, this is outside SCL. There might be additional SCSM specific parameters, if the redundancy issue is taken up in the stack mapping. If not, private P parameters for example at access points might be introduced, if necessary. Because they are private, the redundancy based on them may not work between IEDs of different manufacturers. A typical example is an Ethernet ring based on switches. It provides redundancy against the failure of one switch in the ring, it is however not seen within an SCD file.
- **Application level:** this shall be modelled in SCL. A typical example is the main 1 and main 2 protection IED. Each IED instance providing application redundancy is explicitly modelled having its own name, and any explicitly provided additional communication subnetworks are also modelled in the SCD file. Any coordination between redundant functions is done between the logical nodes which implement the function.

7 SCL description file types

SCL files are used to exchange the configuration data between different tools, possibly from different manufacturers. As already mentioned in Clause 5 (see also Figure 1), there are at least four different purposes for SCL data exchange, and therefore four kinds of SCL files to be distinguished for the data exchange between tools. This is done by means of different file extensions. Nevertheless, the contents of each file shall obey the rules of the Substation Configuration Language SCL defined in the next section. Each file should contain a version and revision number to distinguish different versions of the same file. This means that each tool has to keep the version and revision number information of the last file exported, or read back the last existing file to find out its version.

NOTE The version identifies versions of the SCL file, not versions of the data models used within the tools. This is a private issue of the tools.

The following types of SCL files are distinguished:

- **Data exchange from the IED configuration tool to the system configuration tool** (corresponding to items b) and c) of Clause 5). This file describes the capabilities of an IED. It shall contain exactly one IED section for the IED whose capabilities are described. The IED name shall be **TEMPLATE**. Furthermore, the file shall contain the needed data type templates inclusive logical node type definitions, and may contain an optional substation section, where the substation name shall be **TEMPLATE**. If a substation **TEMPLATE** is defined, the binding of logical node instances to primary equipment indicates a predefined functionality. Any substation, in which this IED shall be used, must match an appropriate substation topology part (example: a CSWI LN bound to an equipment of type CBR is only allowed to control a circuit breaker; a CILO bound to a line disconnector

implements the interlocking logic for a line disconnector). There might be an optional communication section defining possible default addresses of the IED.

The file extension shall be .ICD for IED Capability Description.

- Data exchange from a system specification tool to the system configuration tool. This file describes the single line diagram of the substation and the required logical nodes. It shall contain a substation description section and the needed data type templates and logical node type definitions. If logical nodes allocated to the Substation section are not already allocated to an IED, the IED name reference (value of *iedName* attribute of the *LNode* element) shall be None. If an LN in the substation section is not bound to an IED and also has no logical node type defined, then only the mandatory part of this LN according to IEC 61850-7-4 is specified. If part of the SA system is already known, this might optionally be contained in IED and Communication sections.

The file extension shall be .SSD for System Specification Description.

- Data exchange from the system configuration tool to IED configuration tools (corresponding to items d) and e) of Clause 5). This file contains all IEDs, a communication configuration section and a substation description section.

The file extension shall be .SCD for Substation Configuration Description.

- Data exchange from the IED configuration tool to the IED. It describes an instantiated IED within a project. The communication section contains the current address of the IED. The substation section related to this IED may be present and then shall have name values assigned according to the project specific names. It is an SCD file, possibly stripped down to what the concerned IED shall know. If a compression method is applied, those according to RFC 1952 shall be preferred.

The file extension shall be .CID for Configured IED Description.

A more formal definition of most restrictions for the given parts is given in the XML schema syntax in Annex E. Observe that not all restrictions on IED name and Substation name mentioned above can be described in the schema. To understand the used schema elements, refer to Clauses 8 and 9. Observe however, that this formal definition is informative only and does not belong to the normative SCL language definition. Observe in addition, that not all restrictions on IED name and Substation name mentioned above can be described in the schema. To understand the used schema elements, refer to Clauses 8 and 9.

An IED which is claimed to implement a server according to the IEC 61850 series shall be accompanied by an ICD file or by a tool capable of generating an ICD file, and shall be able to consume an SCD file respectively be accompanied by a tool which can consume the SCD file to configure the communication part of the IED from this SCD file, within the limits declared in the ICD file.

8 The SCL language

8.1 Specification method

The SCL language is based on XML (see Clause 2).

Its syntax definition is described as an W3C XML schema. The remaining Clauses define the appropriate XML schema for SCL and explain its usage in text, enhanced by appropriate (incomplete) examples illustrating the use of the specific features defined, and by additional written requirements, restrictions, and relations to the object model, which shall be used or checked by the application reading or building an SCL file. The complete normative XML schema definition is contained in Annex A. It also contains the formal definitions of those constraints which are easily formulated in XML schema. Constraints on the object model which are not or not easily formulatable in XML schema are additionally described in the appropriate Clauses.

To keep the syntax compact and extensible, the type feature of XML schema is used where appropriate. This introduces a schema element inheritance structure. The inheritance structure

of the main SCL elements is shown in Figure 4 as a UML diagram. UML diagrams can also show containment relations between SCL elements. It has to be kept in mind that these relations are relations between the SCL language elements, and not between the objects represented by the elements, which are shown in Figure 2. However, it has been attempted to keep the XML element relations as close to the object relations as possible.

The following naming conventions are used within the schema:

- Schema type names start with the small letter t (for example *tSubstation*),
- Attribute group definitions start with the acronym ag (for example *agAuthorization*),
- Attribute names start with a small (lower case) letter (for example *name*),
- Element names start with a capital (upper case) letter (for example *Substation*).

Nearly all SCL elements are derived from the *tBaseElement* base type (see for example Figure 4), which allows to add *Private* sections and a descriptive *Text* to the element. It also allows to add additional sub-elements and attributes from other namespaces (other than the target namespace <http://www.iec.ch/61850/2003/SCL>) – such elements must however appear first among all sub-elements. This allows for easy (private) extensions of the model.

Based on *tBaseElement* are the next level of element types:

- *tUnNaming* adds an optional description attribute *desc*
- *tNaming* adds the optional description attribute *desc* and a mandatory name attribute *name*
- *tIDNaming* adds the description attribute *desc* and a mandatory identifier attribute *id*.

In all the previous types, *desc* is a XML normalizedString, i.e., a string that does not contain any carriage return, line feed, or tab character. Its default value is the empty string. Attributes *name* and *id* are both of type *tName*, i.e., also strings that do not contain any carriage return, line feed, or tab character, but cannot be empty.

The resulting inheritance relations for the power system related objects is shown in the UML diagram of Figure 4. Due to this inheritance, also of attributes or of attribute groups, not all attributes are directly defined at an element definition. Nevertheless the description in the following Clauses also describes the inherited attributes, possibly with a reference to a previous description.

For better segmentation and reuse, the whole SCL schema is split in several files containing type definitions (see Table 1).

Table 1 – The files composing the XML schema definition for SCL

File name	Description
SCL_Enums.xsd	The used XML schema enumerations
SCL_BaseSimpleTypes.xsd	The basic simple types used by the other parts
SCL_BaseTypes.xsd	The basic complex type definitions used by the other parts
SCL_Substation.xsd	The Substation related syntax definitions
SCL_Communication.xsd	The Communication related syntax definitions
SCL_IED.xsd	The IED related syntax definitions
SCL_DataTypeTemplates.xsd	The data type template related syntax definitions
SCL.xsd	The main SCL schema syntax definition, which defines the root element of each SCL file.

In the following schema definition Clauses it is assumed that the SCL schema definition file starts as follows:

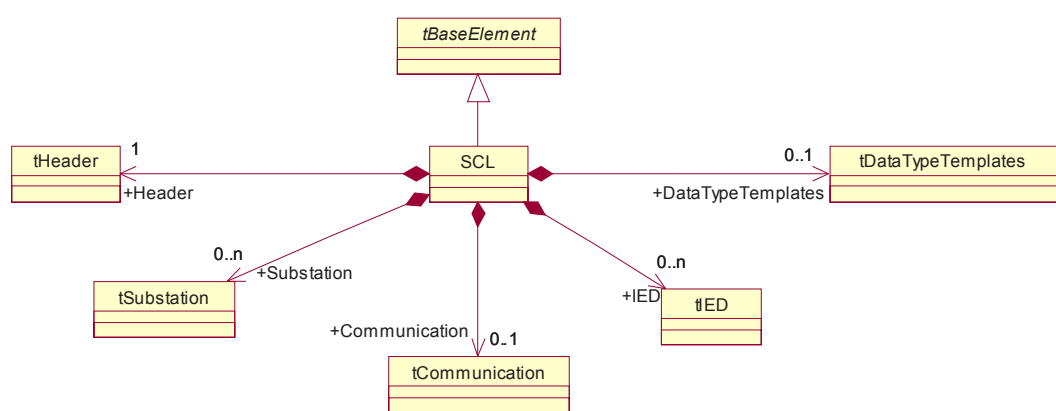
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL"
  xmlns:scl="http://www.iec.ch/61850/2003/SCL"
  xmlns="http://www.iec.ch/61850/2003/SCL"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  finalDefault="extension" version="n.n">
```

where n.n states the SCL version, which is 1.0 for this standard. The schema then ends with

```
</xs:schema>
```

This schema part is not repeated in the following Clauses and Subclauses. For a complete schema definition containing the contents of all above files, see Annex A.

The UML diagram given in Figure 4 gives an overview of how the SCL schema is structured.



IEC 198/04

Figure 4 – UML diagram overview of SCL schema

The basic SCL element is derived from a *tBaseElement* schema type, which allows to contain for example Private and Text definitions. Furthermore, the SCL element shall contain one Header element of type *tHeader*, and may contain Substation elements of type *tSubstation*, a Communication section of type *tCommunication*, IED elements of type *tIED*, and a DataTypeTemplates section of type *tDataTypeTemplates*. All these element types are then handled in later Clauses.

In some cases, the data format of values is important. Wherever possible, the schema defines the data type and therefore also its coding (lexical presentation). But even in cases where this is not possible, the data type coding of XML Schema shall be used. If not explicitly expressed, all element values are XML Schema *strings*, and all attribute values are of the XML schema type *normalizedString*, i.e. they are not allowed to contain tab, carriage return and line feed characters. Further restrictions may be stated either in this part of IEC 61850 or in other parts of the IEC 61850 series, mostly IEC 61850-7-x, IEC 61850-8-x and IEC 61850-9-x. If any XML schema data type is used, it is referenced with the prefix *xs:*, for example *xs:decimal* for decimal number coding. For convenience, an overview about coding of the most types used in SCL is given in Table 42.

8.2 SCL language extensions

8.2.1 General

The core SCL as defined here is designed for a specific purpose described in Clause 5. It can however be used with smaller or bigger extensions such as additional attributes for additional (engineering) tasks. Furthermore, it leaves some communication stack dependent definitions to the SCSMs. Therefore, Subclauses 8.2.2 to 8.2.7 describe SCL extension possibilities.

8.2.2 Data model extensions

Extensions of the data model with semantically new LNs and DOs are covered by the rules stated in IEC 61850-7-x for extensions, and by the SCL approach as a meta language to the data model, i.e. data model element identifications do not appear in the language syntax itself. The name scope of logical node classes, DATA and CDC attributes are described in SCL by stating the appropriate name space values within the appropriate DATA attributes. If additional base data types are needed, then this has to be defined as a schema extension.

8.2.3 Additional semantics to existing syntax elements

Some language elements of SCL such as *desc* and *Text* have a weakly defined semantic, which can be extended by some application. Some elements such as the parameter element *P* have been left open on purpose. An SCSM shall define (additional) semantics to these elements. This is done by defining a *type* value for a *P* parameter with an own semantic.

8.2.4 Data type constraints

The usage of XML schema based data types on the syntactic level already allows the further restriction of the range of some values. A restriction shall use one of the allowed subtypes of the types defined in this core language.

8.2.5 XML name spaces

For all tag elements, (sub-)tags and attributes can be added. These shall however belong to a defined XML name space with defined semantics for all these elements. The used name spaces shall be defined at the main tag (SCL). This namespace should not be the same as the target namespace of the SCL schema (see below). For private name spaces, the used internal name space abbreviation shall start with the character *e*. An example of a standard extension for single line or communication diagram layouts is given in Annex C. The name space URI of this version of the core SCL, which shall be used as default name space in all SCL files, is:

```
xmlns:scl="http://www.iec.ch/61850/2003/SCL"
```

All tools, which comply with this part of IEC 61850, shall be able to import an SCL file with name space definitions, and at least interpret the core SCL as the default name space. Name spaces other than the SCL core, which are not understood by the tool, shall be ignored by it. This especially means that an IED tool which exports data of its own XML name space to an ICD file, can not expect that this information is contained respectively preserved in a SCD file coming from the system configurator tool or another manufacturers IED tool.

NOTE 1 The SCL schema is built in such a way that if the private namespaces are specified in the header but the corresponding schemas are unknown, an XML validator is still able to correctly validate the file (for the parts that are not defined in the SCL schema, the validator will typically only check that they are well-formed).

NOTE 2 The SCL schema demands that elements from private name spaces appear in an SCL file before the elements defined in the SCL schema.

8.2.6 Private parts

For small extensions either by a manufacturer or for a specific project the *Private* parts can be used. The advantage of private parts is that the data content is preserved at data exchange between tools.

Private data entities appear on several levels of the SCL. The contents of these XML elements is, as seen from the SCL, transparent text. If the private part contains XML data, then this has to use an explicit name space, which cannot be the SCL name space. The *Private* element allows also to reference other files by means of a URL at its *source* attribute.

The handling within tools shall be as follows:

The private data is owned by a tool respective by a tool category (for example, a picture generator). The owner is allowed to modify its contents, and normally is the only one able to interpret the data. All other tools, which read private data, have to preserve (store) its contents on SCL import, and regenerate it at the same place if an SCL file containing this part is produced/exported.

Private data for different purposes shall be distinguished by the value of its *type* attribute. If manufacturers use it, this type attribute value should start with a manufacturer-specific string part.

The Private elements have the schema type tPrivate, which is defined as follows:

```
<xs:complexType name="tPrivate" mixed="true">
  <xs:annotation>
    <xs:documentation xml:lang="en"> Allows an unrestricted mixture of character content, element content and
    attributes from any namespace other than the target namespace, along with an optional Type attribute.
  </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="true">
    <xs:extension base="tAnyContentFromOtherNamespace">
      <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="source" type="xs:anyURI" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The attributes of the Private element are defined in Table 2.

Table 2 – Attributes of the Private element

Attribute name	Meaning, usage
type	Distinguishes different (private) purposes of the element contents. The manufacturer or tool name shall be included into the type to be sure it is unique.
source	URL to some file, which contains the private information; only the URL is preserved by the processing tool, not its contents (this stays where it is and has to be preserved with means outside the tool responsibility).

8.2.7 Another XML syntax

A completely new standardised or private XML based syntax for another XML file may be used to extend the SCL data model with additional objects or attributes. In this case, references to the objects contained in the SCL model shall be defined in this new XML file, and the naming philosophy of this part of IEC 61850 shall be followed to be able to identify the objects. The *source* attribute of a *Private* element can be used to link to such additional XML files.

8.2.8 Summary: Standard conformance for extension handling

A tool claiming conformance with this part of IEC 61850 shall as a minimum handle any extensions as follows:

- Import and export the SCL core syntax as a default XML name space; understand all parts of the core syntax referring to the capabilities of the handled IEDs and the intended functionality of the tool.
- Keep all data in private sections and all text elements from import to export (except if modified on purpose within the tool). Keep all data of IEDs, which are not handled, if an SCD file is exported.
- Accept syntactically correct XML name space extensions on import without error message, even if the corresponding contents is ignored.

8.2.9 Extension example

The following extract of an SCL file shows how extensions based on private XML name space can be used for additional XML attributes, additional elements, and for XML elements within the data part of a *Private* element.

```
<?xml version="1.0"?>
<!-- Augmented example file with:
      - Private element
      - using extensions from other namespaces
-->
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL SCL.xsd" xmlns:ext="http://www.private.org">
  <Header id="SCL Example T1-1" nameStructure="IEDName"/>
  <Substation name="baden220_132" ext:myAttribute="my extension attribute">
    <ext:MyElement>This is my extension element</ext:MyElement>
    <Private ext:hello="bla bla">This is my private element <ext:dummy>with sub-elements</ext:dummy> and
a privately defined attribute</Private>
    <PowerTransformer name="T1" type="PTR">
```

Observe that all elements (above the *MyElement*) from other name spaces (*ext* above) than the default SCL name space must come before any SCL elements.

8.3 General structure

An SCL – XML document starts with the XML element *prolog*, and then continues with elements as defined later. The *prolog* shall contain the identification of the XML version and the character coding used. UTF-8 coding is the preferred coding. The whole SCL definition part is contained in the SCL element:

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL SCL.xsd">

  <!-- here come the Header/Substation/IED/Communication/DataTypeTemplate
sections as defined in Clause 9-->
</SCL>
```

where **SCL.xsd** gives the concrete file containing the SCL schema definition.

Note that, for an XML processor, this assumes that the SCL schema definition (i.e., the files enumerated in Table 1) is in the same directory as the SCL instance file. If this is not the case, the full path to the schema must be given here. Alternatively, most XML processors allow you to provide the location of the schemas manually (outside the instance document).

The SCL element shall contain a header section, and at least one of the following sections: Substation, Communication, IED, DataTypeTemplates, which are further explained below. The Substation and the IED sections may appear more than once. Figure 4 gives an overview as an UML diagram. Here is the appropriate XML schema definition.

```
<xs:element name="SCL">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="tBaseElement">
        <xs:sequence>
          <xs:element name="Header" type="tHeader">
            <xs:unique name="uniqueHitem">
              <xs:selector xpath="/scl:History/scl:Hitem"/>
              <xs:field xpath="@version"/>
              <xs:field xpath="@revision"/>
            </xs:unique>
          </xs:element>
          <xs:element ref="Substation" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Communication" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```

<xs:element ref="IED" minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="DataTypeTemplates" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

All elements are derived from the *tBaseElement* type, and therefore inherit the options to contain *Text* and *Private* elements as well as the capability of containing elements and attributes from other name spaces. The elements derived from its sub-types *tUnNaming*, *tNaming*, and *tIDNaming* additionally inherit the *desc* attribute.

8.4 Object and signal designation

The SCL model allows two kinds of object designation:

- 1) A technical key, which is used on engineering drawings and for signal identifications. This is contained in the attribute *name* as identification of each object. If this value is used as reference to an object, it is contained in an attribute name starting with a string denoting the reference target object type, and ending with the string "*Name*". The technical key is used within SCL for referencing other objects. Observe that name is a relative identification within a hierarchy of objects.
- 2) A user oriented textual designation. This is contained in attribute *desc*. Attributes are not allowed to contain carriage return, line feed or tab characters. The semantics of *desc* shall also be relative within an object hierarchy.

Furthermore, a general description tag *Text* can be used to add descriptive textual data. The meaning of this data is not specified further on purpose. Each tool shall preserve imported text data for export.

8.4.1 Object designations in an object hierarchy

In case of the hierarchically structured objects of the substation structure and the product structure, both *name* and *desc* attributes for each object contain only that part which identifies the object within this level of the hierarchy. The full object reference is a pathname and consists of the concatenation of all name parts of higher hierarchy levels up to this level. It is up to the configuring engineer to ensure that the references are unique after concatenation. This shall be reached by using a designation (syntax) convention as specified in IEC 61346-1. This especially means that names of all levels can be directly concatenated to a path name, if the higher level name ends with a number and the lower level name starts with an alpha character or else an intervening character, preferably a dot (.), shall be put between them. If the name within a level is the empty string, no delimiting character is needed at this level. Other separation characters may be specified for name mapping in SCSMs or according to IEC 61346-1. Beneath the mandatory usage of IEC 61346-1 for name syntax, it is strongly recommended to use the whole series for the derivation of functional and IED product names as technical keys. In this case, it should be observed that the special IEC 61346 separator characters like =, +, – shall not appear within SCL names. Only the dot (.) is allowed if names are substructured.

Transition objects, i.e. objects appearing in more than one hierarchical structure, may be identified by several references, one in each structure. In the case of SCL, this applies especially to logical nodes, which are found in the substation functional structure as well as in the IED product structure. There might be other transition points between different structures, but their modelling is outside the scope of SCL.

8.4.2 Signal identifications to be used in the communication system

According to IEC 61850-7-x, signal identifications are built from the following parts (see Figure 5):

- a) A user defined part identifying the logical device LD in the process (LDName).

- b) A (function related) part to distinguish several LNs of the same class within the same IED/LD (LN-Prefix).
- c) The standardised LN class name and the LN instance number, which distinguishes several LNs of the same class and prefix within the same IED/LD.
- d) A signal identification inside a LN consisting of data and attribute name as defined in IEC 61850-7-3 and IEC 61850-7-4.

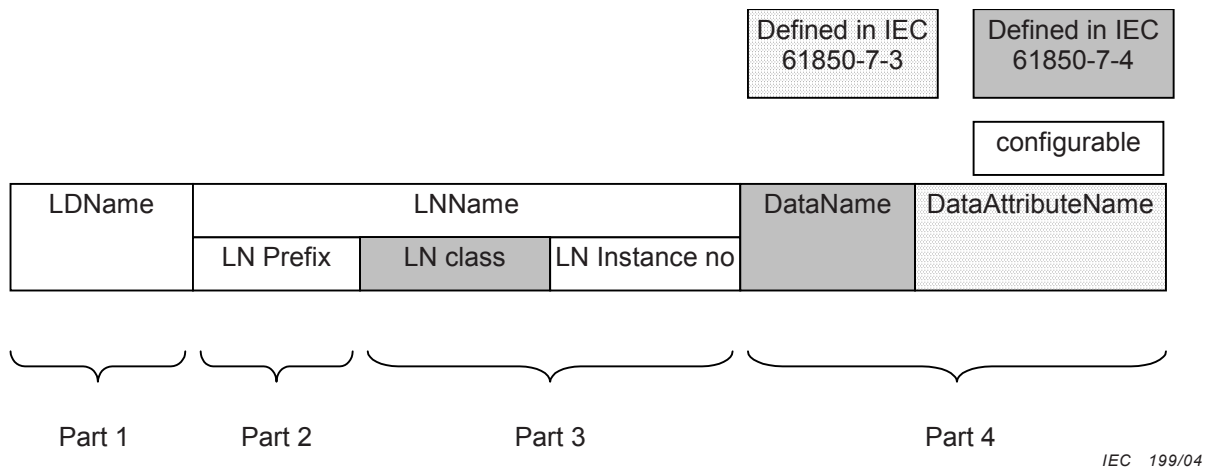


Figure 5 – Elements of the signal identification as defined in IEC 61850-7-2

The name parts 2 and 3 in Figure 5 together form the LN name and distinguish different LN instances within the same LD of an IED. Both can be used freely. A function related LN Prefix is preferably used during functional engineering, or to bind an instantiated LN on an IED to some process semantics. The LN instance number of the name part 3 shall be used to distinguish instantiated LNs, which are not (already) bound to a process semantic (for example a CSWI which is not bound to some specific switch type, prefix=""), or which have the same non-empty prefix.

The mapping of these signal name parts to actual signal names is stack and mapping related and therefore contained in IEC 61850-8-x and IEC 61850-9-x. From the SCL point of view, it is sufficient to determine the contents of these parts for a specific SA system. However, IEC 61850-8-x and IEC 61850-9-x may contain further restrictions on length and contents of name parts.

The DataTypeTemplates definition section of the SCL and the standardised names as defined in IEC 61850-7-3 and IEC 61850-7-4 determine the possible values for name parts 3 and 4 in Figure 5. The LN instance number and the prefix are defined in the IED section of the SCL.

For name parts 1 and 2 in Figure 5 there are two options (see also Figure 6 and Figure 7). For both, a separation of part 1 in Figure 5 into an IEDName and a LD instance name LDInst within this IED is used.

- 1) **Function related naming:** part 1 in Figure 5 is the name of the object of the substation section, to which the LN is attached. If it is a PrimaryDevice, use the name parts from substation name to bay name as part 1, and use the PrimaryDevice name (possibly followed by a subequipment name) as part 2. Concatenate the IED LD Inst to part 1. If LNs are attached to higher levels than the bay level, naturally the part 1 has to be shortened appropriately, and the part 2 in Figure 5 stays empty, or can be used for the level where the LN is attached to.

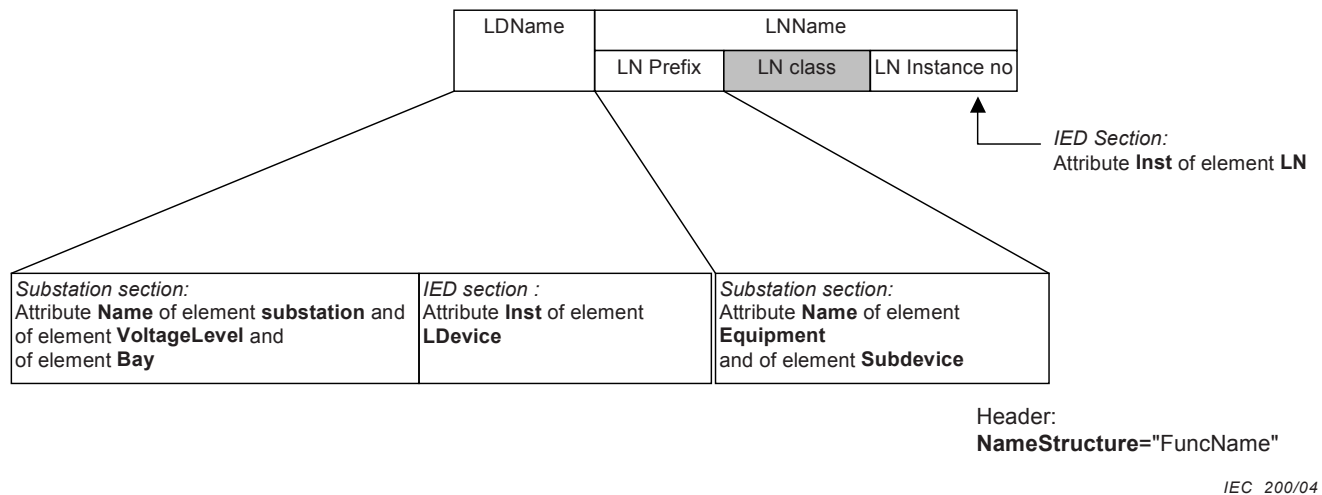


Figure 6 – Elements of the signal name using functional naming

2) **Product related naming:** Part 1 in Figure 5 is the name of the IED in the IED (product) section, on which the LN is configured, concatenated with the LD Instance number. Part 2 stays as predefined within the IED (see Figure 7).

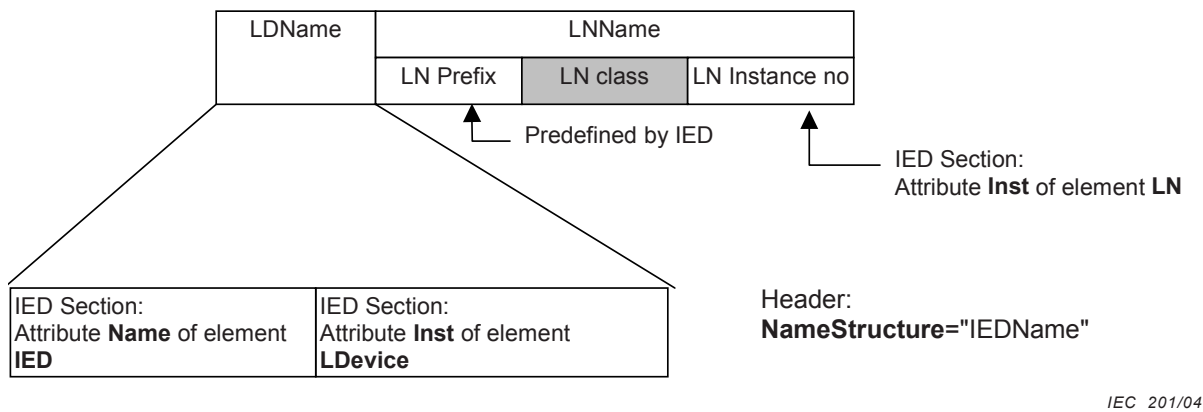
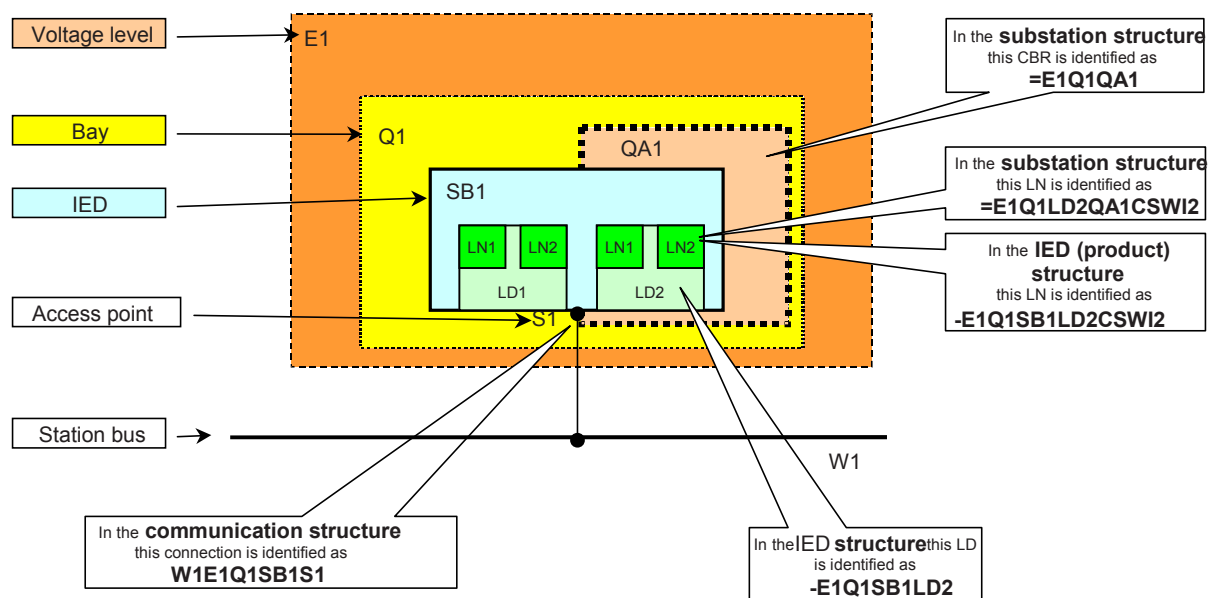


Figure 7 – Elements of the signal name using product naming

The SCL model leaves both options open, but allows the header part to specify if option 1 (functional naming) or option 2 (product naming) is taken for signal naming at communication time. It is recommended to use the LN instance number in such a way that the LN class and LN instance number together are always unique. This allows the way of naming (with/without prefix) to be changed at a later time, and even to later replace preconfigured prefixes by prefixes related to the functional structure. The use of these options might however be restricted in case that an IED has fixed prefix and LN instance number, that is it does not allow to change this for a certain LN instance later on. In this case only product related naming can be chosen.

8.4.3 Naming example

Figure 8 shows an example of an IED with LNs, which control a circuit breaker QA1 of bay Q1 in voltage level E1. The naming is chosen according to the IEC 61346 series. In this example, the IED as a product has the same higher-level product designation part according to the bay (-E1Q1) as the controlled circuit breaker QA1 has in its functional designation (=E1Q1QA1). Figure 8 shows the resulting references within different structures, and the resulting LN reference for communication.



IEC 202/04

Figure 8 – Names within different structures of the object model

If DATA of LN2 of LN class CSWI within LD2 are now named with names from the function structure, then the LN reference according to IEC 61850-7-2 would be E1Q1LD2/QA1CSWI2. If the reference were taken from the product structure, it would be E1Q1SB1LD2/CSWI2. Observe that the whole name in each case shall be unique within the system, which is the case for both names above. However, in the case of the functional name, the LD reference E1Q1LD2 alone is **not necessarily unique within the system** (only within the IED), because there could be another IED within bay E1Q1 with an LD2. Only the relation of E1Q1QA1CSWI2 to the IED E1Q1SB1 in the reference from the Substation structure to the IEDs enables one to find the correct IED for this LD, and E1Q1LD2 then identifies uniquely the LD within this IED.

NOTE If the reference is taken from the functional structure, and if there could be several IEDs within the functional part before the LD name part, it is recommended that the LD instances be identified with names from the functional structure. If, for example, there are protection and control IEDs within the same bay, the LD name part could identify the protection and control subfunctions within the bay.

If a LN prefix is already used at a preconfigured IED, then this will always be taken as name part. In case of functional naming, the engineering process has to assure that the prefix and the device/subdevice identification are identical.

9 The SCL syntax elements

9.1 Header

The header serves to identify an SCL configuration file and its version, and to specify options for the mapping of names to signals. The UML diagram given in Figure 9 gives an overview on its structure.

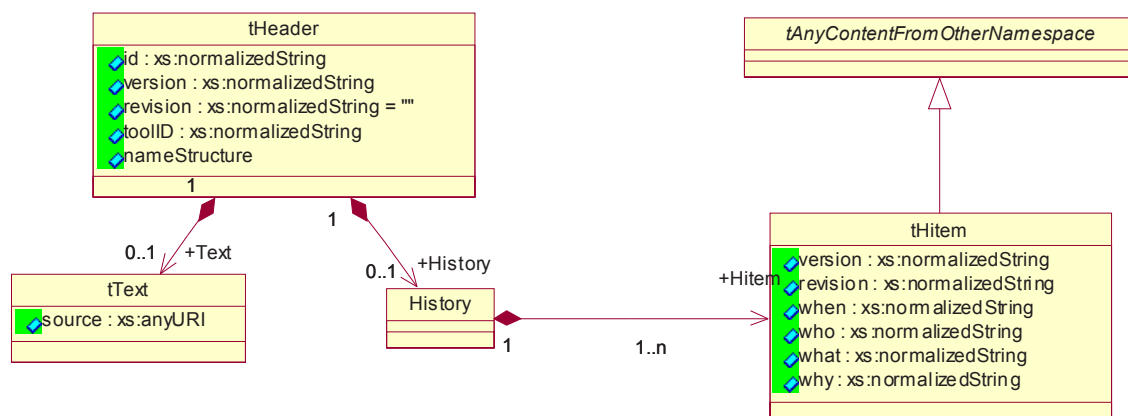


Figure 9 – UML diagram of Header section

IEC 203/04

Here is the XML schema definition part

```

<xs:complexType name="tHeader">
  <xs:sequence>
    <xs:element name="Text" type="tText" minOccurs="0"/>
    <xs:element name="History" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Hitem" type="tHitem" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:normalizedString" use="required"/>
  <xs:attribute name="version" type="xs:normalizedString"/>
  <xs:attribute name="revision" type="xs:normalizedString" default=""/>
  <xs:attribute name="toolID" type="xs:normalizedString"/>
  <xs:attribute name="nameStructure" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:Name">
        <xs:enumeration value="FuncName"/>
        <xs:enumeration value="IEDName"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
  
```

The attributes of the Header element are defined in Table 3.

Table 3 – Attributes of the Header element

Attribute name	Description
id	A string identifying this SCL file, mandatory (may be empty)
version	The version of this SCL configuration file (may be empty)
revision	The revision of this SCL configuration file, by default the empty string meaning the original before any revision
toolID	The manufacturer specific identification of the tool that was used to create the SCL file
nameStructure	Element indicating if communication system signal names are built from the substation function structure (FuncName) or from the IED product structure (IEDName)

The *Text* element is optional, and has the following syntax:

```
<xs:complexType name="tText" mixed="true">
  <xs:annotation>
    <xs:documentation xml:lang="en">Allows an unrestricted mixture of character content and element content
and attributes from any namespace other than the target namespace.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="true">
    <xs:extension base="tAnyContentFromOtherNamespace">
      <xs:attribute name="source" type="xs:anyURI" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Instead of putting text into this element, a reference to another file can also be given as URI in the *source* attribute.

NOTE The *Text* syntax element for describing text is used in several places, essentially in all elements derived from the *tBaseElement* (see 8.1 and Clause A.1).

The revision history is optional. The same syntax can be used also for other documents requiring a revision history. If present, it should have the following form:

```
<xs:complexType name="tHitem" mixed="true">
  <xs:annotation>
    <xs:documentation xml:lang="en"> Allows an unrestricted mixture of character content and element content
and attributes from any namespace other than the target namespace, along with the 6 following attributes: Version,
Revision, When, Who, What, and Why</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="true">
    <xs:extension base="tAnyContentFromOtherNamespace">
      <xs:attribute name="version" type="xs:normalizedString" use="required"/>
      <xs:attribute name="revision" type="xs:normalizedString" use="required"/>
      <xs:attribute name="when" type="xs:normalizedString" use="required"/>
      <xs:attribute name="who" type="xs:normalizedString"/>
      <xs:attribute name="what" type="xs:normalizedString"/>
      <xs:attribute name="why" type="xs:normalizedString"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The history contains several history item entries. Each item identifies a (previously) approved version of this SCL file. A text within the items can be used to explain further details to this version.

Table 4 – Attributes of the History item (Hitem) element

Attribute name	Description
version	The version of this history entry
revision	The revision of this history entry
when	Date when the version/revision was released
who	Who made/approved this version/revision
what	What has been changed since the last approval
why	Why the change has happened

The following example shows a completely filled header example without history, where the signal names are taken from the substation function structure:

```
<Header id="1KHL1000546" version="1" revision=""
  toolId="mySystemTool V1.2" nameStructure="FuncName">My SA Project
</Header>
```

9.2 Substation description

The substation section serves to describe the functional structure of a substation, and to identify the primary devices and their electrical connections. For an industrial process or to describe whole power networks, it is possible to have several substation sections, one for each substation served by the SAS. By means of logical nodes attached to the primary system elements, this section defines additionally the SA system functionality (for example, in an SSD file), or, in the case where the logical nodes are already allocated to IEDs (SCD file), the relation of IED functions to the power system.

Note that the *name* attribute is always mandatory and shall not be the empty string. If the substation section is used as template within an ICD file, then the name shall be TEMPLATE. The name value is also a global identification of the substation, because it shall be unique for all substations contained in the SCL file.

If the *desc* attribute is missing, its default value is an empty string.

Logical nodes (LNode) can be attached at each level of the structure (i.e., substation, voltage level, bay, equipment, subequipment respective function, subfunction). Power transformers (*PowerTransformer*) can also be attached at the structure levels substation, voltage level and bay. Conducting equipments (*ConductingEquipment*) can only be attached to the bay level. Logical node instances at the same level shall have different identifications.

The UML diagram given in Figure 10 gives an overview on the substation section:

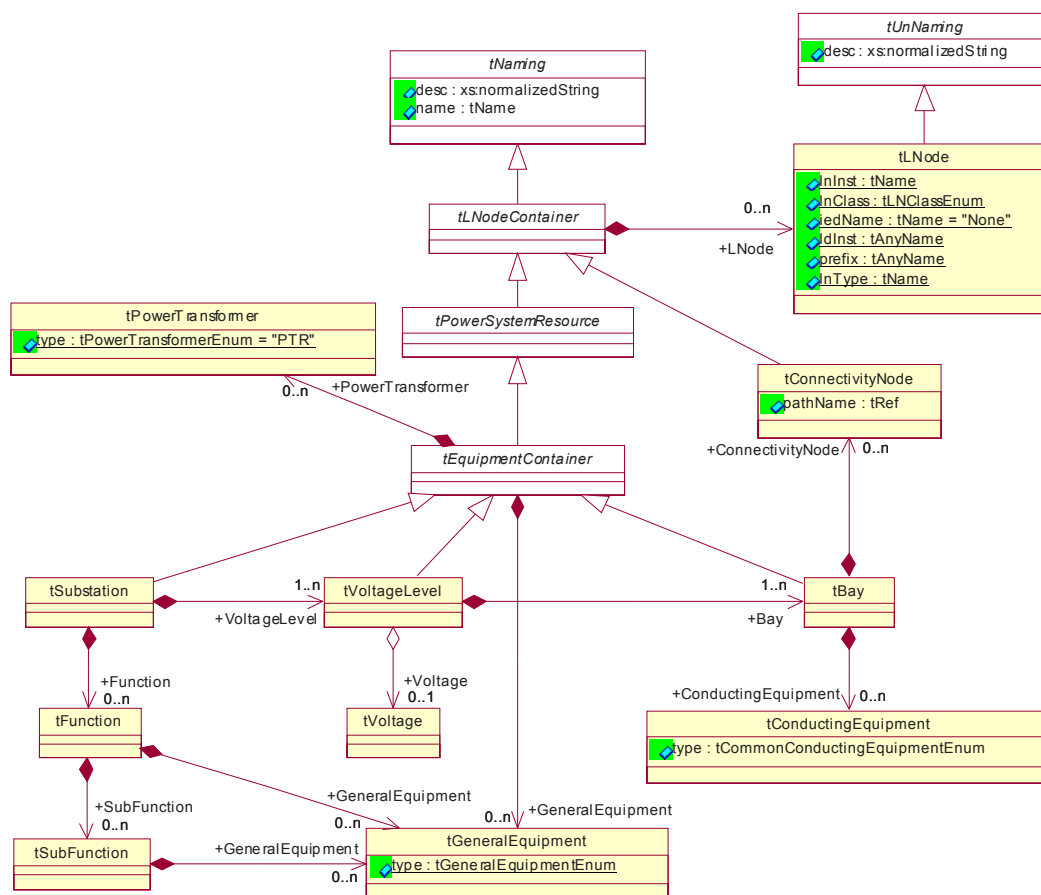


Figure 10 – UML diagram of Substation section

The appropriate schema part is as follows:

These basic type definitions are used for the elements:

```

<xs:include schemaLocation="SCL_BaseTypes.xsd"/>
<xs:attributeGroup name="agVirtual">
  <xs:attribute name="virtual" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
<xs:complexType name="tLNodeContainer" abstract="true">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="LNode" type="tLNode" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tPowerSystemResource" abstract="true">
  <xs:complexContent>
    <xs:extension base="tLNodeContainer"/>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tEquipmentContainer" abstract="true">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:sequence>
        <xs:element name="PowerTransformer" type="tPowerTransformer" minOccurs="0"
maxOccurs="unbounded">
          <xs:unique name="uniqueWindingInPowerTransformer">
            <xs:selector xpath="/scl:TransformerWinding"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="GeneralEquipment" type="tGeneralEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tEquipment" abstract="true">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAbstractConductingEquipment" abstract="true">
  <xs:complexContent>
    <xs:extension base="tEquipment">
      <xs:sequence>
        <xs:element name="Terminal" type="tTerminal" minOccurs="0" maxOccurs="2"/>
        <xs:element name="SubEquipment" type="tSubEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tConductingEquipment">
  <xs:complexContent>
    <xs:extension base="tAbstractConductingEquipment">
      <xs:attribute name="type" type="tCommonConductingEquipmentEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSubEquipment">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attribute name="phase" type="tPhaseEnum" use="optional" default="none"/>
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Then the Substation type is as follows:

```
<xs:complexType name="tSubstation">
  <xs:complexContent>
    <xs:extension base="tEquipmentContainer">
      <xs:sequence>
        <xs:element name="VoltageLevel" type="tVoltageLevel" maxOccurs="unbounded">
          <xs:unique name="uniqueBayInVoltageLevel">
            <xs:selector xpath="/scl:Bay"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniquePowerTransformerInVoltageLevel">
            <xs:selector xpath="/scl:PowerTransformer"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInVoltageLevel">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueChildNameInVoltageLevel">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="Function" type="tFunction" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueSubFunctionInFunction">
            <xs:selector xpath="/scl:SubFunction"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInFunction">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The Substation element is of type *tSubstation* as shown above. It is an *tEquipmentContainer*, i.e. it might contain logical nodes (*LNode*) as well as power transformers (*PowerTransformer*). Further it contains at least one voltage level, and optionally several *Function* elements. System functions or equipment, which do not belong to the power system, can be described by the *Function* element.

The general *Substation* element (of type *tSubstation*), which is referred to by the *SCL* element, includes additionally several identity constraints:

- Within a *Substation*, there cannot be two *VoltageLevel* elements with the same *name*.
- Within a *Substation*, there cannot be two *PowerTransformer* elements with the same *name*.
- Within a *Substation*, there cannot be two *Function* elements with the same *name*.
- Within a *Substation*, there cannot be two *LNode* elements with the same combination of *lnInst*, *lnClass*, *iedName*, *ldInst*, and *prefix*.
- Further, in order to avoid any ambiguities, within a *Substation* there cannot be two direct child elements with the same name.

Restrictions

- The substation name shall be unique within an *SCL* file.
- For a primary system template within an *ICD* file, the substation name shall be *TEMPLATE*. There can be a maximum of one substation template in one *SCL* file.

- Within a *Substation*, the attribute *pathName* of a *ConnectivityNode* acts as a key (a *ConnectivityNode* may appear at bay level below the *Substation*). This implies that there cannot be two *ConnectivityNode* elements with the same *pathName*. The *connectivityNode* attribute of each *Terminal* in this *Substation* must then refer to one of these keys.

9.2.1 Voltage level

A *VoltageLevel* element is of type *tVoltageLevel* as shown below. It has an optional element *Voltage* of type *tVoltage*, which can be used to state the voltage of this voltage level. Furthermore, as *tEquipmentContainer* it might contain logical nodes (*LNode*), *GeneralEquipment* and power transformers (*PowerTransformer*), and it contains one or several bays by means of the *Bay* element.

```
<xs:complexType name="tVoltageLevel">
  <xs:complexContent>
    <xs:extension base="tEquipmentContainer">
      <xs:sequence>
        <xs:element name="Voltage" type="tVoltage" minOccurs="0"/>
        <xs:element name="Bay" type="tBay" maxOccurs="unbounded">
          <xs:unique name="uniquePowerTransformerInBay">
            <xs:selector xpath="/scl:PowerTransformer"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueConductingEquipmentInBay">
            <xs:selector xpath="/scl:ConductingEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInBay">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueChildNameInBay">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Several identity constraints are defined (in fact, they are defined in *tSubstation* above):

- Within a *VoltageLevel*, there cannot be two *Bay* with the same *name*.
- Within a *VoltageLevel*, there cannot be two direct child *PowerTransformer* elements with the same *name*.
- Within a *VoltageLevel*, there cannot be two direct child *GeneralEquipment* with the same *name*.
- Further, in order to avoid any ambiguities, within a *VoltageLevel*, there cannot be two direct child elements with the same *name*.

Restrictions

- The voltage level name shall be unique within the substation.
- The bay name shall be unique within a voltage level.

9.2.2 Bay level

The *Bay* element is of type *tBay*. As an equipment container, it might contain power transformers, general equipment and logical nodes. Additionally, it might host conducting equipment (*ConductingEquipment*) and connectivity nodes (*ConnectivityNode*), which are used to define topological connections between equipment within a single line diagram.

```

<xs:complexType name="tBay">
  <xs:complexContent>
    <xs:extension base="tEquipmentContainer">
      <xs:sequence>
        <xs:element name="ConductingEquipment" type="tConductingEquipment" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ConnectivityNode" type="tConnectivityNode" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The *ConnectivityNode* element allows the explicit definition of connectivity nodes within this bay, and as *tLNodeContainer*, logical nodes (*LNode*) can be attached to it. Its *Text* sub-element can be used to contain some freely usable description. Its *name* attribute identifies the *ConnectivityNode* instance within the bay; its *pathName* is an absolute reference within the SCL file. The pathname is build by all higher level references down to the connectivity nodes name, concatenated with the character "/". For instance, if the connectivity node L1 is within bay Q2 of voltage level E1 of substation Baden, then the pathname is "Baden/E1/Q2/L1".

NOTE 1 The separator "/" has been purposely selected, because the dot "." might appear as part of the names at higher hierarchy levels, for example at bay level.

```

<xs:complexType name="tConnectivityNode">
  <xs:complexContent>
    <xs:extension base="tLNodeContainer">
      <xs:attribute name="pathName" type="tRef" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

NOTE 2 If a bus bar bay does not contain any primary devices, it can be modelled as a bay that contains only connectivity nodes.

Several identity constraints are defined (in fact, they are defined in *tVoltageLevel* – see the code in Clause 9.2.1):

- Within a *Bay*, there cannot be two direct child *PowerTransformer* elements with the same *name*.
- Within a *Bay*, there cannot be two direct child *ConductingEquipment* elements with the same *name*.
- Within a *Bay*, there cannot be two direct child *GeneralEquipment* with the same *name*.
- Further, in order to avoid any ambiguities, within a *Bay*, there cannot be two direct child elements with the same *name*.

An example substation section can be found in 9.2.7.

9.2.3 Power equipment

The power equipment is subdivided into the *PowerTransformer* and *ConductingEquipment*. The *PowerTransformer* might appear in each equipment container, and contains the transformer windings as special *ConductingEquipment*. To each transformer winding, a tap changer can be allocated. All other *ConductingEquipment* might appear in the bays only. All equipment is derived from the *tEquipment* base type, and the *ConductingEquipment* from the *tAbstractConductingEquipment* type.

The UML diagram given in Figure 11 gives an overview about the equipment inheritance relations.

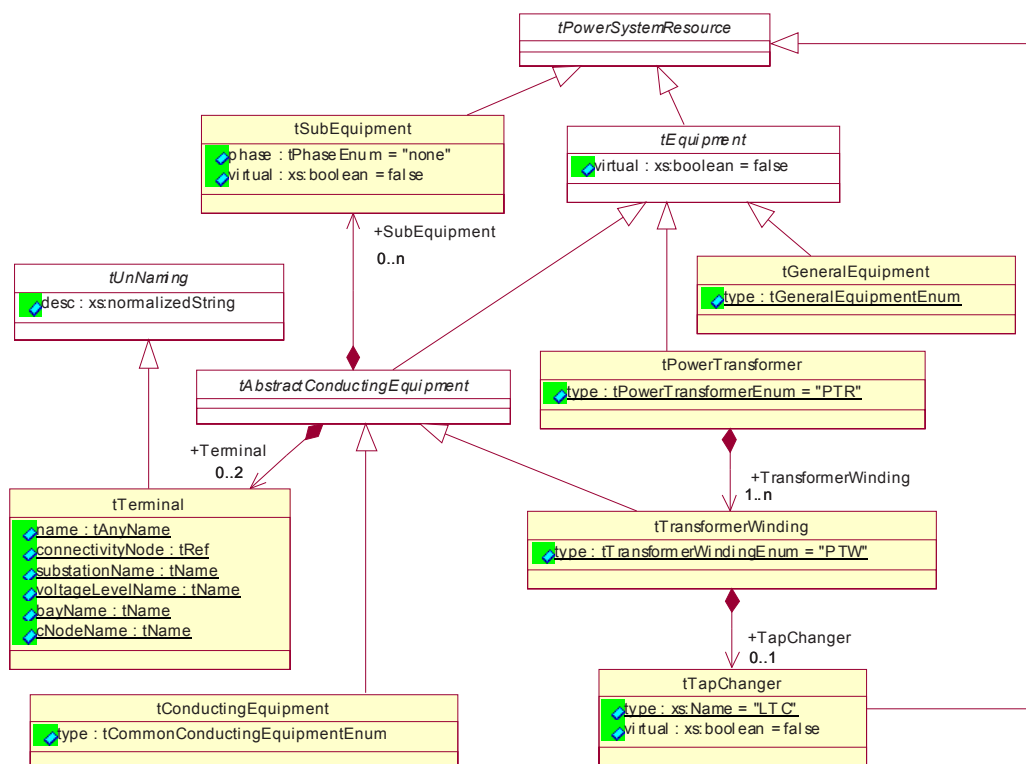


Figure 11 – UML diagram for equipment type inheritance and relations

IEC 205/04

The appropriate schema part is as follows.

```
<xs:complexType name="tEquipment" abstract="true">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAbstractConductingEquipment" abstract="true">
  <xs:complexContent>
    <xs:extension base="tEquipment">
      <xs:sequence>
        <xs:element name="Terminal" type="tTerminal" minOccurs="0" maxOccurs="2"/>
        <xs:element name="SubEquipment" type="tSubEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tConductingEquipment">
  <xs:complexContent>
    <xs:extension base="tAbstractConductingEquipment">
      <xs:attribute name="type" type="tCommonConductingEquipmentEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSubEquipment">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attribute name="phase" type="tPhaseEnum" use="optional" default="none"/>
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="tPowerTransformer">
  <xs:complexContent>
    <xs:extension base="tEquipment">
      <xs:sequence>
        <xs:element name="TransformerWinding" type="tTransformerWinding" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="type" type="tPowerTransformerEnum" use="required" fixed="PTR"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tTransformerWinding">
  <xs:complexContent>
    <xs:extension base="tAbstractConductingEquipment">
      <xs:sequence>
        <xs:element name="TapChanger" type="tTapChanger" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="type" type="tTransformerWindingEnum" use="required" fixed="PTW"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tTapChanger">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attribute name="type" type="xs:Name" use="required" fixed="LTC"/>
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tGeneralEquipment">
  <xs:complexContent>
    <xs:extension base="tEquipment">
      <xs:attribute name="type" type="tGeneralEquipmentEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Observe that all equipment of type *tEquipment*, and all subequipment of type *tSubEquipment* as well as the tapchanger (*tTapChanger*) also have, beneath the normal *name* and *desc* attributes, an optional *virtual* attribute (*agVirtual*). If the substation section is just used for function related naming, this is not really used. However, there are some applications where functions (LNs) calculate values belonging to some 'virtual' equipment, for example a phase current is calculated from the measured values of the other two phases. In this case, it is important to know that the third phase CT is only 'virtually' there, and not so in reality. This can be indicated by setting the *virtual* attribute to true. Its default value is false.

Terminals and their connections to the connectivity nodes (see *tAbstractConductingEquipment*) model the substation topology on the level of a single line, i.e. the number of phases and special connections between phases are not considered here. The maximum number of possible connections to connectivity nodes depends on the terminals available for a device function type. The type codes given in Table 5 for attribute *type* are selected, based as far as possible on IEC 61850-7-4 LN class names.

Table 5 – Primary apparatus device type codes

Type code	Meaning	Number of terminals (connections to different connectivity nodes)
CBR	Circuit Breaker	2
DIS	Disconnect or earthing switch	2
VTR	Voltage Transformer	1
CTR	Current Transformer	2
PTW	Power Transformer Winding	1
PTR	Power Transformer	Implicit via windings
LTC	Load Tap Changer	Part of winding
GEN	Generator	1
CAP	Capacitor bank	1/2
REA	Reactor	1/2
CON	Converter	1/2
MOT	Motor	1
EFN	Earth Fault Neutralizer (Peterson coil)	1
PSH	Power Shunt	2
AXN	Auxiliary Network	none
BAT	Battery	1
BSH	Bushing	2
CAB	Power cable	2
GIL	Gas Insulated Line	2
LIN	Power overhead line or line segment: line segments connected by connectivity nodes form a line. A line segment within a substation could be used to attach for example special LNs, or physical line properties. For a GIS line segment, GIL could be used instead.	2
RRC	Rotating reactive component	1
SAR	Surge arrester	1
TCF	Thyristor controlled frequency converter	2
TCR	Thyristor controlled reactive component	2
IFL	Infeeding line; substation limiting object; models a possibly infeeding power network line outside the substation at the single line border	1

In addition, private types may be used. To allow compatibility with future enhancements of this standard, they shall start with the character E, contain only capital letters, and have at least three letters.

A terminal definition contains the reference to a connectivity node to which the equipment is connected (ConnectivityNode in the model of Figure 2), and optionally the name of the equipment terminal, which connects to this connectivity node. As reference to the ConnectivityNode the path name as well as a list of attributes is used. Both are mandatory. The path name reference allows to check the connection consistency already on XML schema level, while the attribute list is easier to interpret by most tools.

```

<xs:complexType name="tTerminal">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:attribute name="name" type="tAnyName" use="optional"/>
      <xs:attribute name="connectivityNode" type="tRef" use="required"/>
      <xs:attribute name="substationName" type="tName" use="required"/>
      <xs:attribute name="voltageLevelName" type="tName" use="required"/>
      <xs:attribute name="bayName" type="tName" use="required"/>
      <xs:attribute name="cNodeName" type="tName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Table 6 – Attributes of the Terminal element

Attribute name	Description
Name	The optional relative name of the terminal at this Equipment. The default is the empty string, which means that the name of the ConnectivityNode is also the terminal identification.
Desc	Descriptive text to the terminal
ConnectivityNode	The pathname of the connectivity node to which this terminal connects. If the Equipment shall not be connected, then the whole Terminal element shall be removed.
SubstationName	The name of the substation containing the connectivityNode
VoltageLevelName	The name of the voltage level containing the connectivityNode
BayName	The name of the bay containing the connectivityNode
CnodeName	The (relative) name of the connectivityNode within its bay

Equipment terminal identifications are in general only needed if the device polarizes the power flow, i.e. the connections are not interchangeable. If the terminal name attribute is left empty, but a terminal designation needed, then the default is the equipment identification (substationName voltageLevelName bayName equipmentName) together with the connectivity node identification connectivityNode.

There is one predefined connectivity node with the name **grounded**. This is used to model earth potential. Thus, an earthing switch is an isolator (equipment type DIS) that is connected on one side to the connectivity node **grounded**. It is up to the generating tool to decide if **grounded** is one single node for the whole substation, or a separate node at each place where connected, or something in between, for example per bay or voltage level, by generating appropriate pathNames.

9.2.4 SubEquipment level

SubEquipment are parts of the power equipment, like a pump is part of a switch, or like a phase of a switch is a part of the whole switch. They especially allow the specification of a phase relation of LNs. Therefore SCL allows SubEquipment only at Conducting Equipment.

```

<xs:complexType name="tSubEquipment">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:attribute name="phase" type="tPhaseEnum" use="optional" default="none"/>
      <xs:attributeGroup ref="agVirtual"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Table 7 – Attributes of the SubEquipment element

Attribute name	Description
Name	The identification of the subequipment relative to the equipment designation (for example L1, if related to phase A)
Desc	A textual description of the subdevice relative to the device
Phase	The phase to which the subdevice belongs. The following phase values are allowed: <i>A</i> , <i>B</i> , <i>C</i> , <i>N</i> (neutral), <i>all</i> (meaning all three phases), and <i>none</i> (default, meaning not phase related)
Virtual	Set to <i>true</i> , if the subequipment (for example phase CT) does not exist in reality, but its values are just calculated. Optional, default is <i>false</i>

9.2.5 Substation function logical nodes

All equipment and equipment containers are also logical node containers. The logical node (abbreviated here as LN) defines the SA function part performed at the appropriate level of the hierarchy. The LNode element identifies the SA function by specifying a logical node as defined in IEC 61850-5 and IEC 61850-7-x. The optional attribute *desc* may contain some operator-related text describing the LN and its usage.

```
<xs:complexType name="tLNode">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:attribute name="lnInst" type="tAnyName" use="optional" default=""/>
      <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
      <xs:attribute name="iedName" type="tName" use="optional" default="None"/>
      <xs:attribute name="ldInst" type="tAnyName" use="optional" default=""/>
      <xs:attribute name="prefix" type="tAnyName" use="optional" default=""/>
      <xs:attribute name="lnType" type="tName" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The logical node and its function is identified by the element attributes. The *LNode* element can be used within an SSD for functional specification, without allocation to an IED. In this case the *iedName* shall be **None**. For more detailed specification *lnType* may refer to a logical node type definition (0), which then also defines the optional DATA required to exist in this special case, or defines certain values, which some (configuration) parameters shall have. If the logical node is later allocated to an IED within an SCD, then the value of this *lnType* attribute can be ignored, or may be used to check if the logical node type used on the IED fulfills the requirements.

Table 8 – Attributes of the LNode element

Attribute name	Description
lnInst	The LN instance identification. Can only be missing for lnClass=LLN0, meaning as value here the empty string.
lnClass	The LN class as defined in IEC 61850-7-x.
iedName	The name of the IED which contains the LN, none if used for specification (default if attribute is not specified).
ldInst	The LD instance on the IED which contains the LN, empty respectively irrelevant if used for specification.
prefix	The LN prefix used in the IED (if needed; default if not specified is the empty string).
lnType	The logical node type definition containing more detailed functional specification. Might be empty, if the LN is allocated to an IED.

NOTE For LLN0, the value of inst is the empty string. In all other cases, it is an unsigned integer.

The *iedName* identifies the IED on which the LN resides, the *IdInst* the LD within this IED to which the LN belongs. The attributes *prefix*, *InClass* and *inst* (meaning the LN instance identification according to IEC 61850-7-x) then identify the logical node within that LD. In this way, the binding between the substation function and the SA system is defined.

9.2.6 Non power equipment

To be able to model the connection of IED hosted logical nodes to other functions than power system related ones like fire fighting equipment or door supervision, the *Substation* section contains the element *Function*, which again contains an arbitrary number of *SubFunction* elements. Both elements are logical node containers and may also contain *GeneralEquipment*, if necessary. Both *Function* and *Subfunction* have the *name* and *desc* attributes like *Substation* itself, and might also contain the *Text* and *Private* elements. However, there are no connections defined between the equipment.

```
<xs:complexType name="tFunction">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:sequence>
        <xs:element name="SubFunction" type="tSubFunction" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueGeneralEquipmentInSubFunction">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="GeneralEquipment" type="tGeneralEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="tSubFunction">
  <xs:complexContent>
    <xs:extension base="tPowerSystemResource">
      <xs:sequence>
        <xs:element name="GeneralEquipment" type="tGeneralEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The equipment type allowed within Function and Subfunction is termed GeneralEquipment.

```
<xs:complexType name="tGeneralEquipment">
  <xs:complexContent>
    <xs:extension base="tEquipment">
      <xs:attribute name="type" type="tGeneralEquipmentEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

From the equipment type list (Table 5) this is AXN, BAT, MOT. Furthermore, all private codes (containing only capital letters, starting with "E") may be used.

9.2.7 Substation section example

The following SCL example for a system specification SSD contains a substation section for substation baden220_132 with one transformer T1 between voltage levels D1 and E1, and a bay E1Q1.

The transformer T1 has two windings W1 and W2. W1 is connected to a 220 kV voltage level D1 at bay Q1, connectivity node L1. Winding W2 is connected to the bay Q2 in 110 kV voltage level E1. From the attachment of logical nodes in the SSD file it can be seen that there is the measurement of a current transformer at the transformer, and a differential protection. At the 220 kV side (bay D1Q1) there is a distance protection.

The 132kV bay E1Q2 contains a circuit breaker QA1 and a bus bar disconnecter QB1, both electrically connected together at connectivity node L0, as well as voltage transformer U1 at connectivity node L2, and current transformer I1 between the connectivity nodes L1 and L2. The connectivity node within the same bay is explicitly defined. A logical node of type CSWI controls each switch, and the CILO LN handles the interlocking. No association to IEDs is defined, as this is a functional specification only, so the *iedName* is per default None. In addition, the possibility of defining more details by *InType* references has not been used here.

```
<?xml version="1.0"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL
SCL.xsd">
  <Header id="SSD Example " nameStructure="IEDName"/>
  <Substation name="baden220_132">
    <PowerTransformer name="T1" type="PTR">
      <LNode lnInst="1" lnClass="PDIF" lnInst="F1" />
      <LNode lnInst="1" lnClass="TCTR" lnInst="C1" />
      <TransformerWinding name="W1" type="PTW">
        <Terminal connectivityNode="baden220_132/D1/Q1/L1" substationName="baden220_132"
voltageLevelName="D1" bayName="Q1" cNodeName="L1"/>
      </TransformerWinding>
      <TransformerWinding name="W2" type="PTW">
        <Terminal connectivityNode="baden220_132/E1/Q2/L3" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L3"/>
      </TransformerWinding>
    </PowerTransformer>
    <VoltageLevel name="D1">
      <Voltage multiplier="k" unit="V">220</Voltage>
      <Bay name="Q1">
        <LNode lnInst="1" lnClass="PDIS" lnInst="F1" />
        <ConductingEquipment name="I1" type="CTR">
          <Terminal connectivityNode="baden220_132/D1/Q1/L1" substationName="baden220_132"
voltageLevelName="D1" bayName="Q1" cNodeName="L1"/>
        </ConductingEquipment>
        <ConnectivityNode name="L1" pathName="baden220_132/D1/Q1/L1"/>
      </Bay>
    </VoltageLevel>
    <VoltageLevel name="E1">
      <Voltage multiplier="k" unit="V">132</Voltage>
      <Bay name="Q2">
        <ConductingEquipment name="QA1" type="CBR">
          <LNode lnInst="1" lnClass="CILO" lnInst="C1" iedName="D1Q1SB4"/>
          <Terminal connectivityNode="baden220_132/E1/Q2/L1" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L1"/>
          <Terminal connectivityNode="baden220_132/E1/Q2/L2" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L2"/>
        </ConductingEquipment>
        <ConductingEquipment name="QB1" type="DIS">
          <LNode lnInst="2" lnClass="CSWI" lnInst="C1" />
          <LNode lnInst="2" lnClass="CILO" lnInst="C1" />
          <Terminal connectivityNode="baden220_132/E1/W1/B1" substationName="baden220_132"
voltageLevelName="E1" bayName="W1" cNodeName="B1"/>
          <Terminal connectivityNode="baden220_132/E1/Q2/L1" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L1"/>
        </ConductingEquipment>
        <ConductingEquipment name="I1" type="CTR">
          <Terminal connectivityNode="baden220_132/E1/Q2/L2" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L2"/>
          <Terminal connectivityNode="baden220_132/E1/Q2/L3" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L3"/>
        </ConductingEquipment>
        <ConductingEquipment name="U1" type="VTR">
          <Terminal connectivityNode="baden220_132/E1/Q2/L3" substationName="baden220_132"
voltageLevelName="E1" bayName="Q2" cNodeName="L3"/>
        </ConductingEquipment>
      </Bay>
    </VoltageLevel>
  </Substation>
</SCL>
```

```

voltageLevelName="E1" bayName="Q2" cNodeName="L3"/>
  </ConductingEquipment>
  <ConnectivityNode name="L1" pathName="baden220_132/E1/Q2/L1"/>
  <ConnectivityNode name="L2" pathName="baden220_132/E1/Q2/L2"/>
  <ConnectivityNode name="L3" pathName="baden220_132/E1/Q2/L3"/>
</Bay>
<Bay name="W1">
  <ConnectivityNode name="B1" pathName="baden220_132/E1/W1/B1"/>
</Bay>
</VoltageLevel>
</Substation>
</SCL>

```

9.3 IED description

9.3.1 General

The IED section describes the (pre-)configuration of an IED: its access points, the logical devices, and logical nodes instantiated on it. Furthermore, it defines the capabilities of an IED in terms of communication services offered and, together with its LNTYPE, instantiated data (DO) and its default or configuration values. There shall be one IED section for each IED. IED names (name attribute) shall be unique within the file. If only the descriptions of pre-configured IEDs are contained in the file, the name shall be **TEMPLATE** to indicate that the IED has not been bound to a place in the project. The system configurator tool should handle this as an IED type, i.e. a pre – configured product type, from which an arbitrary number of product (hardware) instances can be produced.

NOTE Because the IED *name* is unique within a system, it is also usable as a reference.

A special IED *Router* function is introduced. An IED containing a router function connects different subnetworks by means of all its access points. The router IED may have no logical devices and no logical nodes. In this case, it is managed and supervised by a separate network management system, beyond the scope of this standard. A router is a limiting border, which real time related message types cannot cross. These message types are:

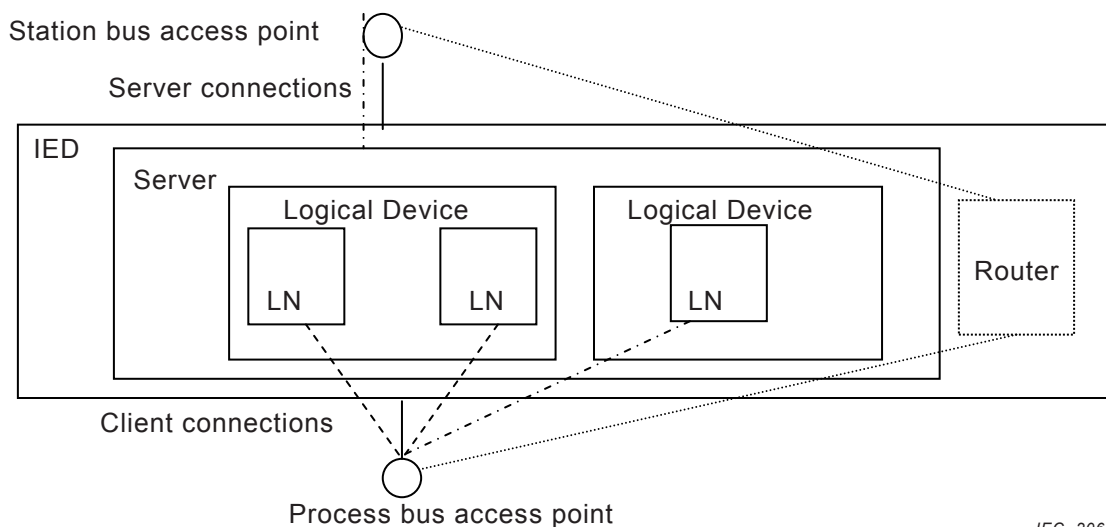
- time synchronisation messages,
- GSE messages,
- sampled measurement values.

All other messages are routed through with some time delay.

In addition to the stand-alone router IED described above, the router function can reside on an IED containing additionally clients or servers.

An access point may belong to a server with logical devices, which contain logical nodes. In this case, the server of the access point provides access to the LDs and LNs, while the LNs as clients may use all IED access points (not only those of the server) to access data (on LNs on servers) on other IEDs. An access point always needs a server, if the IED shall be supervised remotely, because the LN0 and LPHD of the server's logical device are used to supervise and control the IED. Only if all LNs on an IED use an access point as a client only, and the IED is not supervised, may an IED without a server be used.

It is recommended that an IED contains at least one server. An access point without a server may then be used to get data from 'lower level' busses, i.e. a bay unit from process bus. However, this data from the lower level bus cannot be seen directly on the higher level bus unless a router function also resides on this IED. Figure 12 gives a typical example of an IED connected to station bus and process bus.



IEC 206/04

Figure 12 – IED structure and access points

By means of the short address feature, it is possible to define a translation of logical names to short addresses on a data attribute basis.

The usage and meaning of short addresses may be defined in an SCSM (stack mapping). In this case, the system configuration tool handles them. If an SCSM does not define this, the IED tools might use short address related attributes as reference to IED internal addresses. In this case, the IED tool handles them. All other tools shall just import and reexport their contents.

More about short addresses is found in 9.5.4.3.

Figures 13 to 15 give an overview of the IED related schema part in the form of UML diagrams.

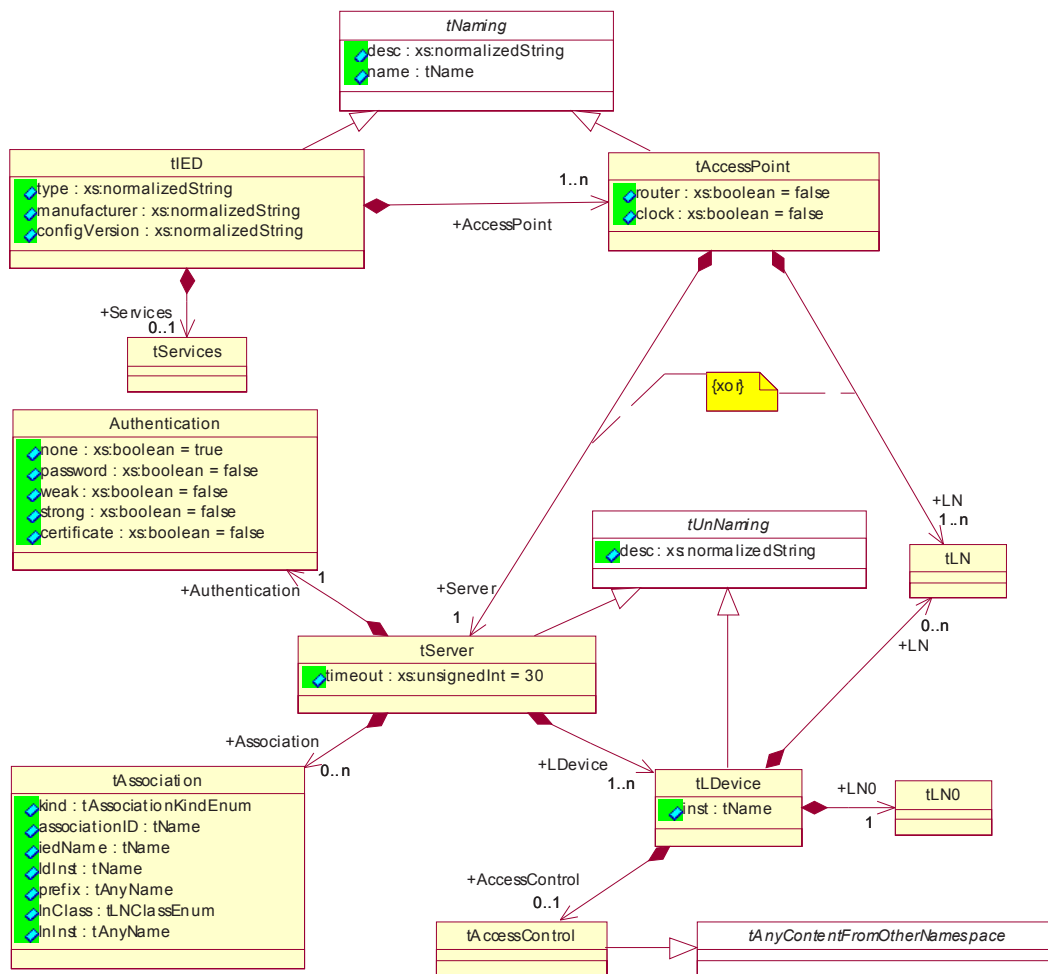


Figure 13 – UML description of IED related schema part – base

IEC 207/04