INTERNATIONAL STANDARD

IEC 61850-6

First edition
2004-03

Communication networks and systems in substations –

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

Reference number
IEC 61850-6:2004(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (http://www.iec.ch/searchpub/cur_fut.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (http://www.iec.ch/online_news/justpub/jp_entry.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 61850-6

First edition
2004-03

Communication networks and systems in substations –

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

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XG

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMUNICATION NETWORKS AND SYSTEMS IN SUBSTATIONS –

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

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61850-6 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

The text of this standard is based on the following documents:

FDIS	Report on voting
57/693/FDIS	57/713/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IEC 61850 consists of the following parts, under the general title *Communication networks and systems in substations:*

Part 1: Introduction and overview

Part 2: Glossary

Part 3: General requirements

Part 4: System and project management

Part 5: Communication requirements for functions and device models

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

Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models

Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)

Part 7-3: Basic communication structure for substation and feeder equipment – Common data classes

Part 7-4: Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes

Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

Part 9-1: Specific Communication Service Mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link

Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3

Part 10: Conformance testing[1]

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this standard may be issued at a later date.

---

[1] Under consideration.

INTRODUCTION

This part of IEC 61850 specifies a description language for the configuration of electrical substation IEDs. This language is called Substation Configuration description Language (SCL). It is used to describe IED configurations and communication systems according to IEC 61850-5 and IEC 61850-7-x. It allows the formal description of the relations between the substation automation system and the substation (switchyard). At the application level, the switchyard topology itself and the relation of the switchyard structure to the SAS functions (logical nodes) configured on the IEDs can be described.

SCL allows the description of an IED configuration to be passed to a communication and application system engineering tool, and to pass back the whole system configuration description to the IED configuration tool in a compatible way. Its main purpose is to allow the interoperable exchange of communication system configuration data between an IED configuration tool and a system configuration tool from different manufacturers.

IEC 61850-8-x and IEC 61850-9-x, which concern the mapping of IEC 61850-7-x to specific communication stacks, may extend these definitions according to their need with additional parts, or just by restrictions on the way the values of objects have to be used.

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) 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.

b) 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.

c) Pre-configured IEDs with a pre-configured semantic for a process part of a certain structure, for example a double busbar GIS line feeder.

d) 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.

e) 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.

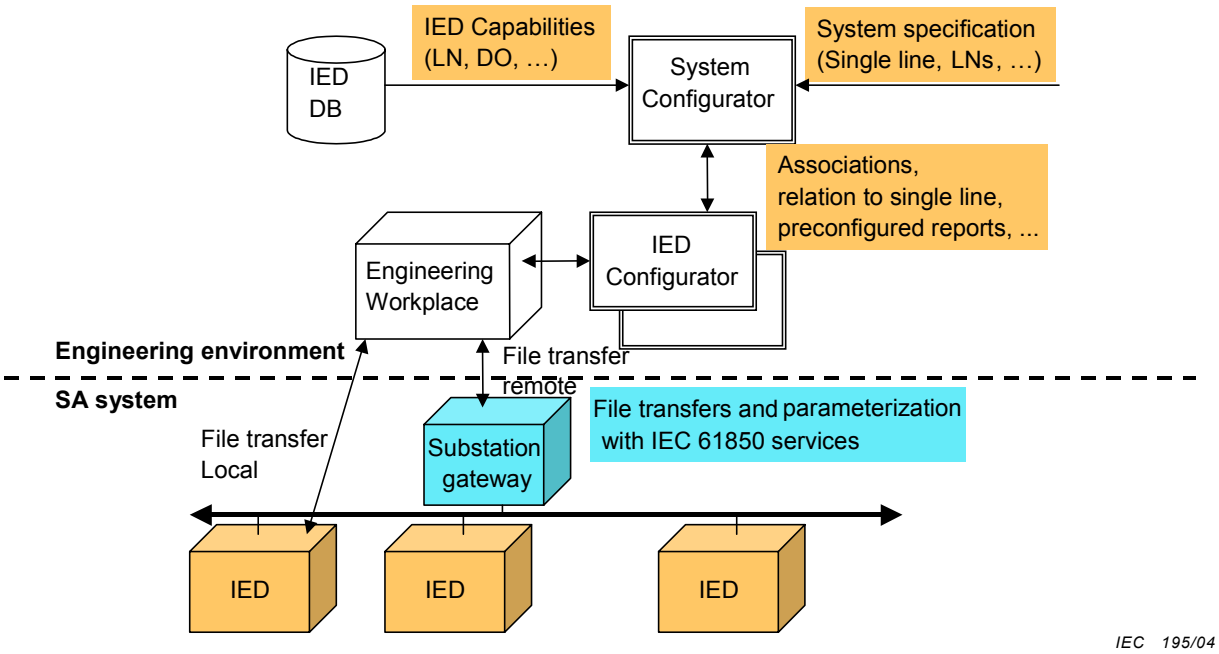


Figure 1 – Reference model for information flow in the configuration process

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.

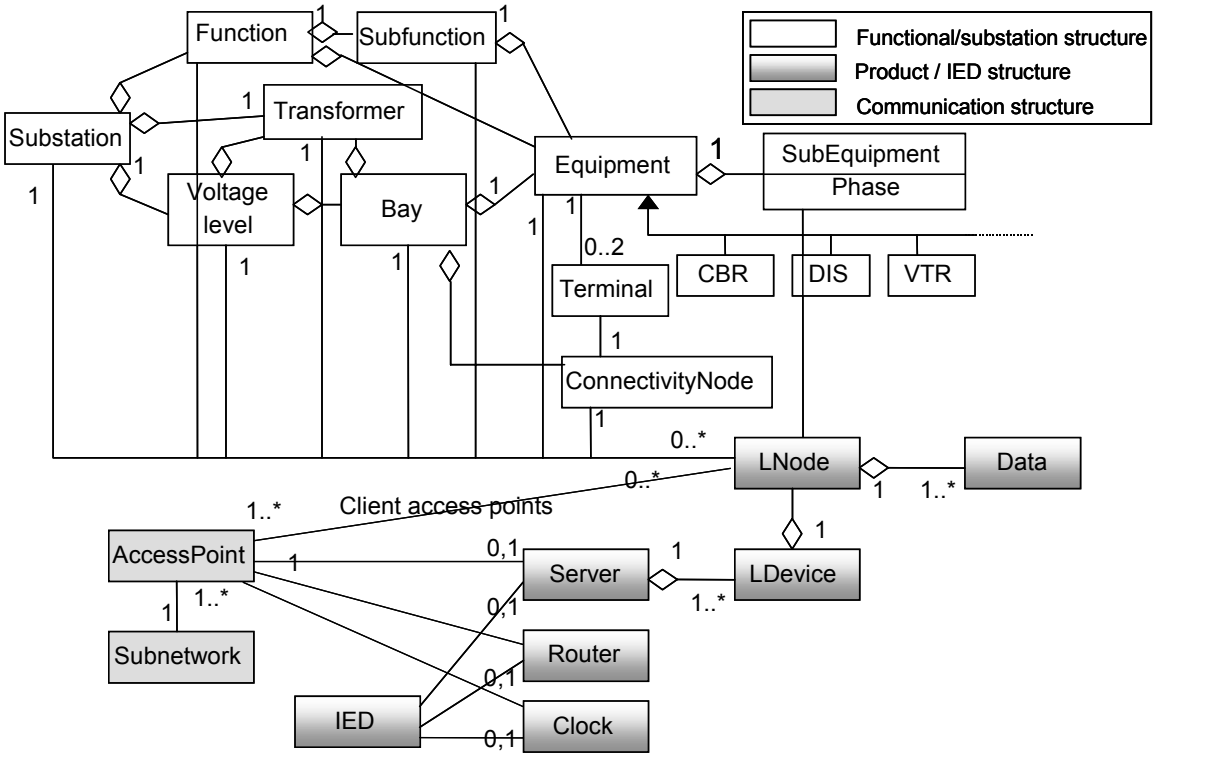


Figure 2 – SCL object model

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.

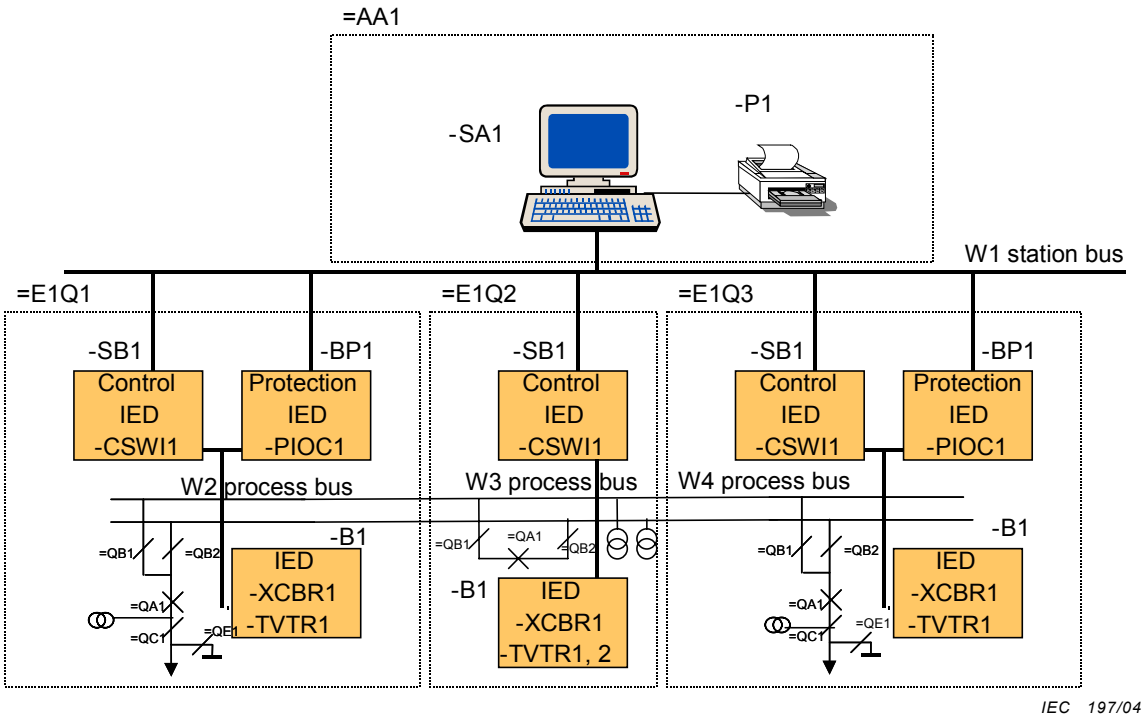


Figure 3 – Configuration example

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 respective 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 IED which

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.

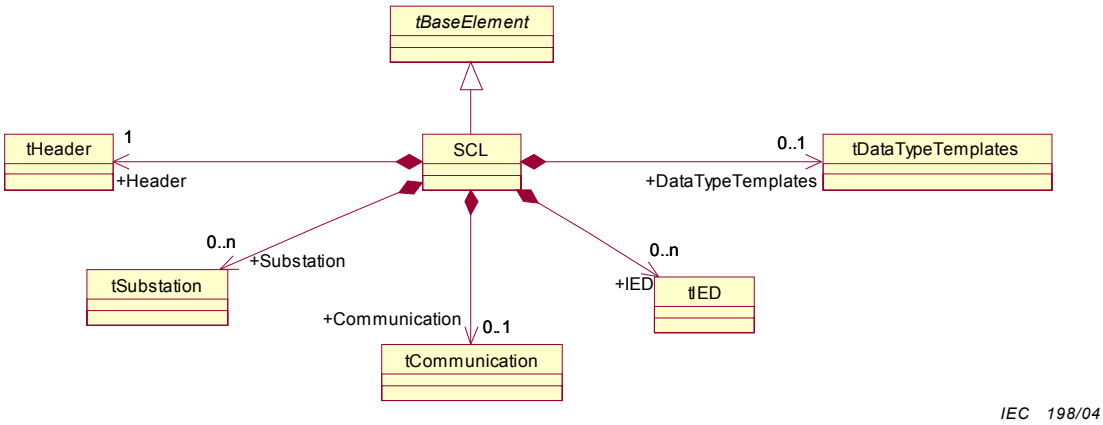


Figure 4 – UML diagram overview of SCL schema

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 *string*s, 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: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.

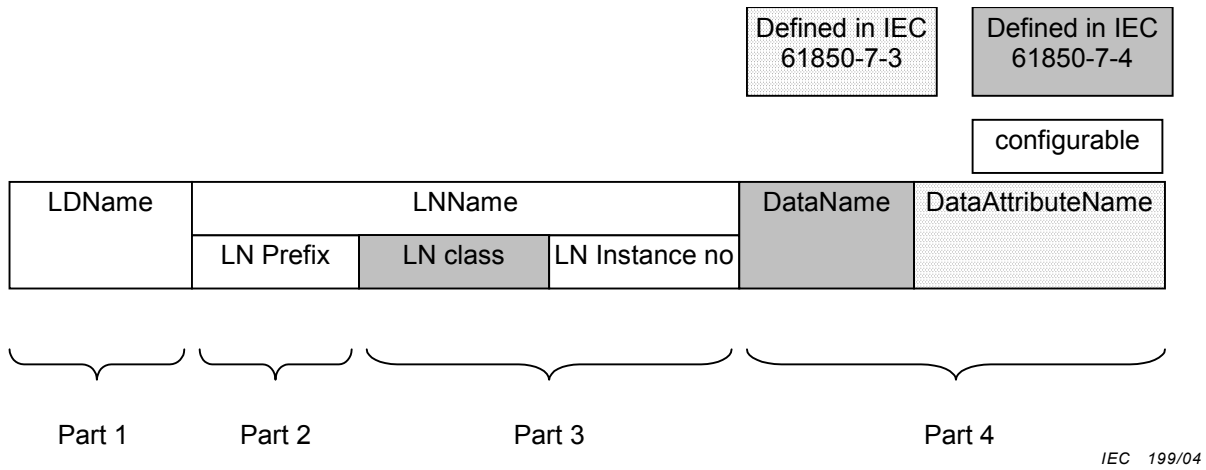


Figure 5 – Elements of the signal identification as defined in IEC 61850-7-2

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.

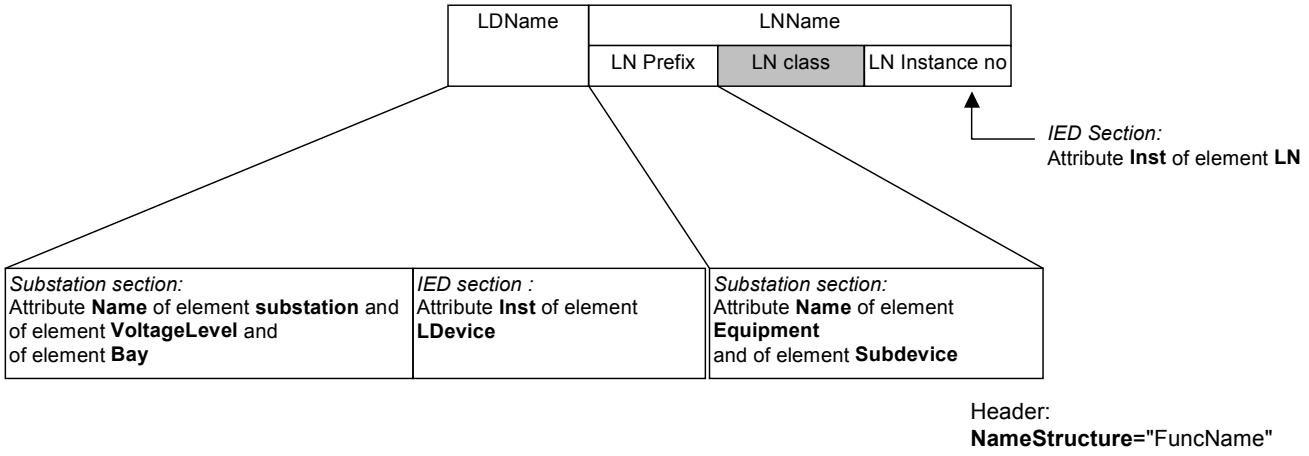


Figure 6 – Elements of the signal name using functional naming

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).

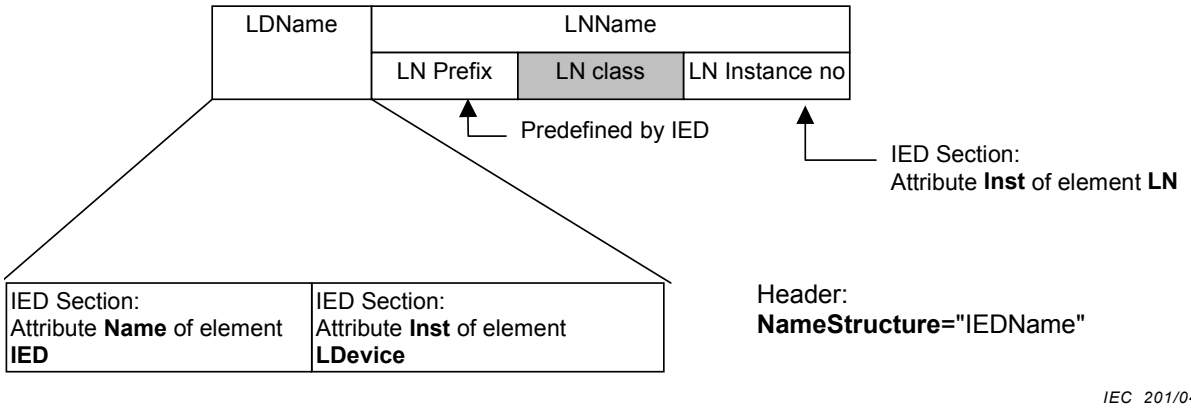


Figure 7 – Elements of the signal name using product naming

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.

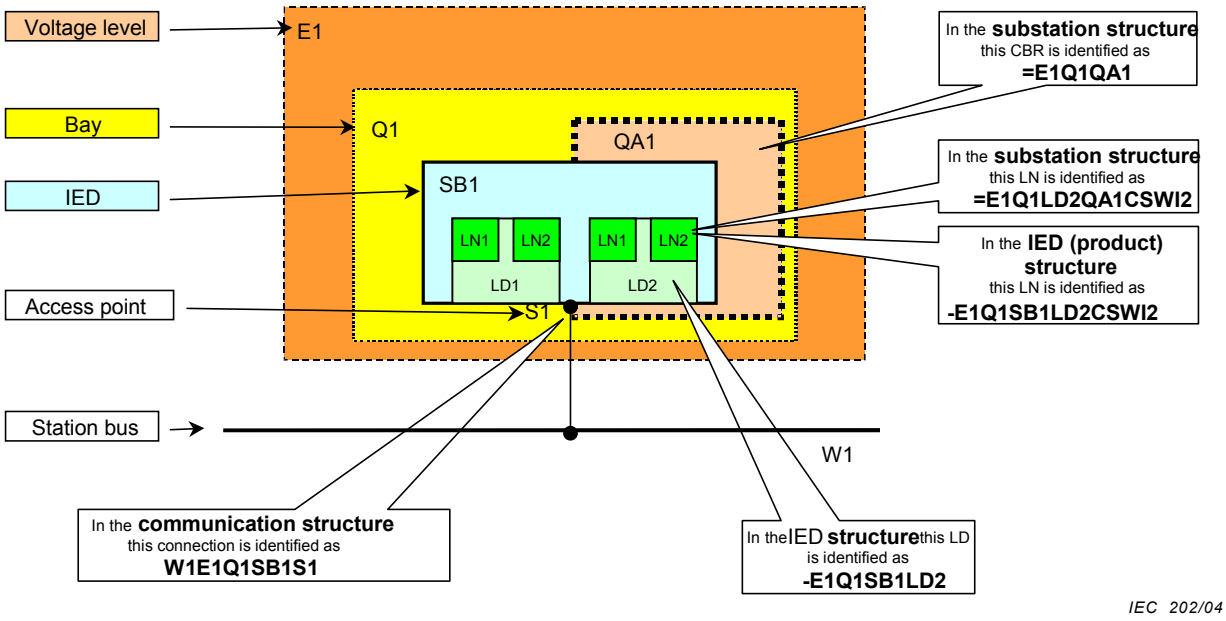


Figure 8 – Names within different structures of the object model

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 instancess 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.

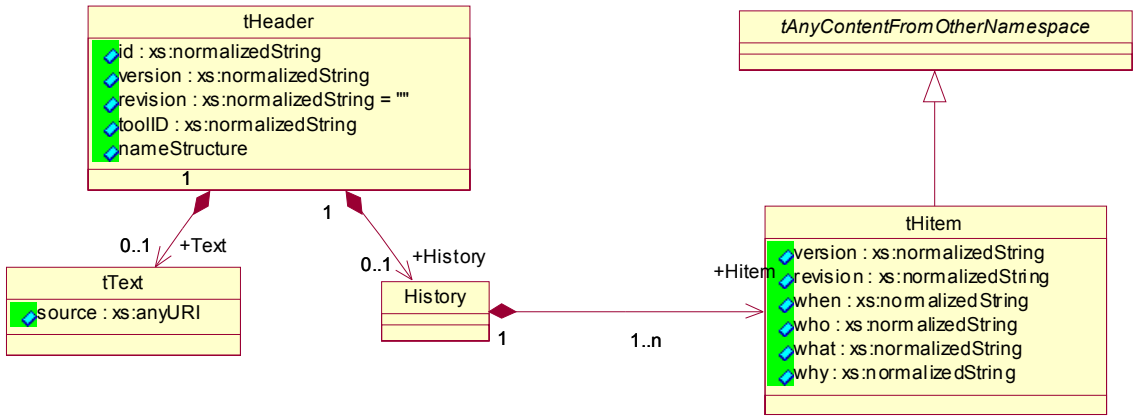


Figure 9 – UML diagram of Header section

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:

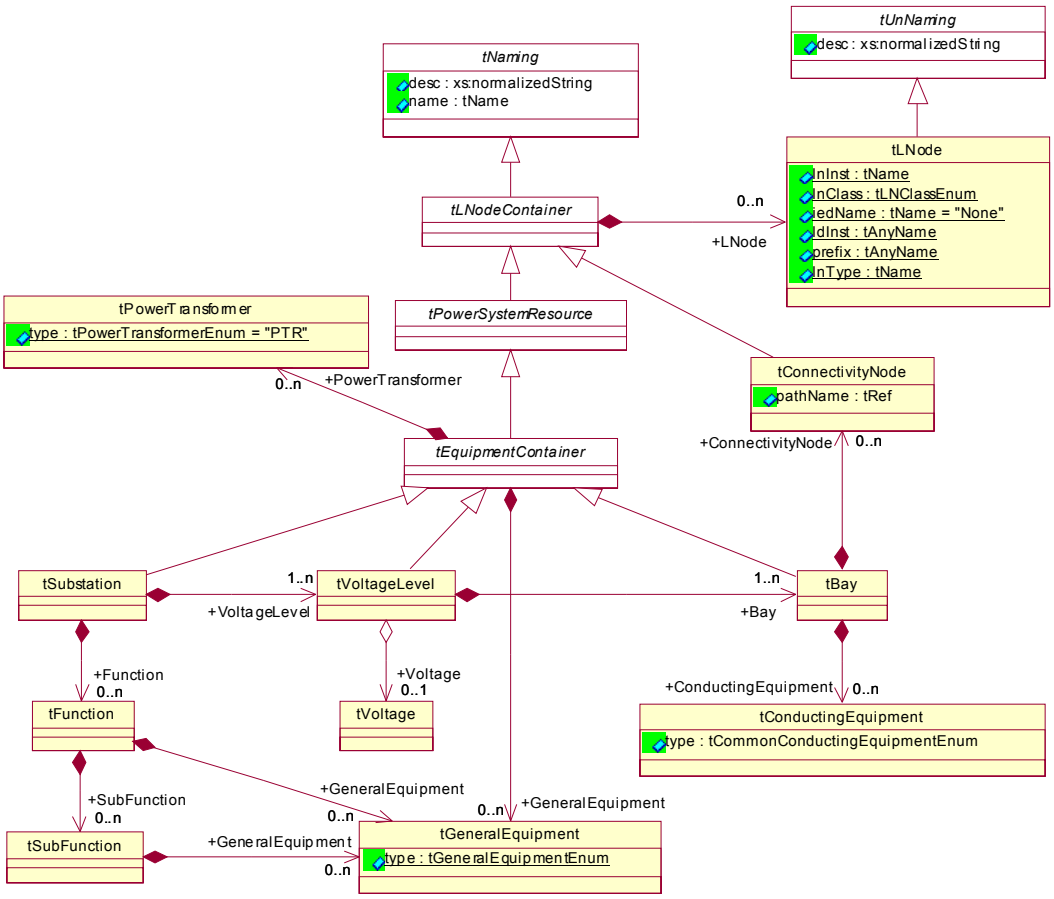


Figure 10 – UML diagram of Substation section

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>
```

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):

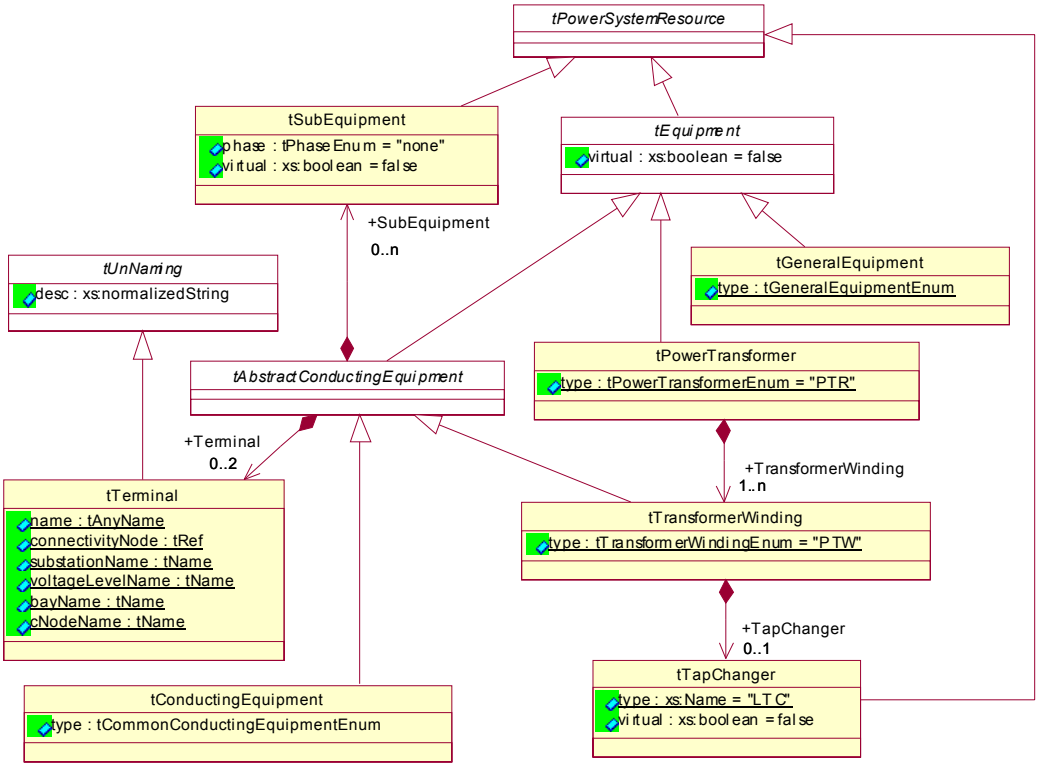
- Within a *Bay*, there cannot be two direct child *PowerTransformer* elements with the same *name*.
- Within a *Bay*, there cannot be two direct child *ConductingEquipment* elements with the same *name*.
- Within a *Bay*, there cannot be two direct child *GeneralEquipment* with the same *name*.
- Further, 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 *ConductingEqupipment*. 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.



IEC 205/04

Figure 11 – UML diagram for equipment type inheritance and relations

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: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	Disconnector 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 arrestor	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: *A, B, C, N* (neutral), *all* (meaning all three phases), and *none* (default, meaning not phase related)
Virtual	Set to *true*, if the subequipment (for example phase CT) does not exist in reality, but its values are just calculated. Optional, default is *false*

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 ldInst the LD within this IED to which the LN belongs. The attributes *prefix*, *lnClass* 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 disconnector 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 explicitely 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 *lnType* 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" ldInst="F1" />
            <LNode lnInst="1" lnClass="TCTR" ldInst="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" ldInst="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" ldInst="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" ldInst="C1" />
                    <LNode lnInst="2" lnClass="CILO" ldInst="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>
                <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.

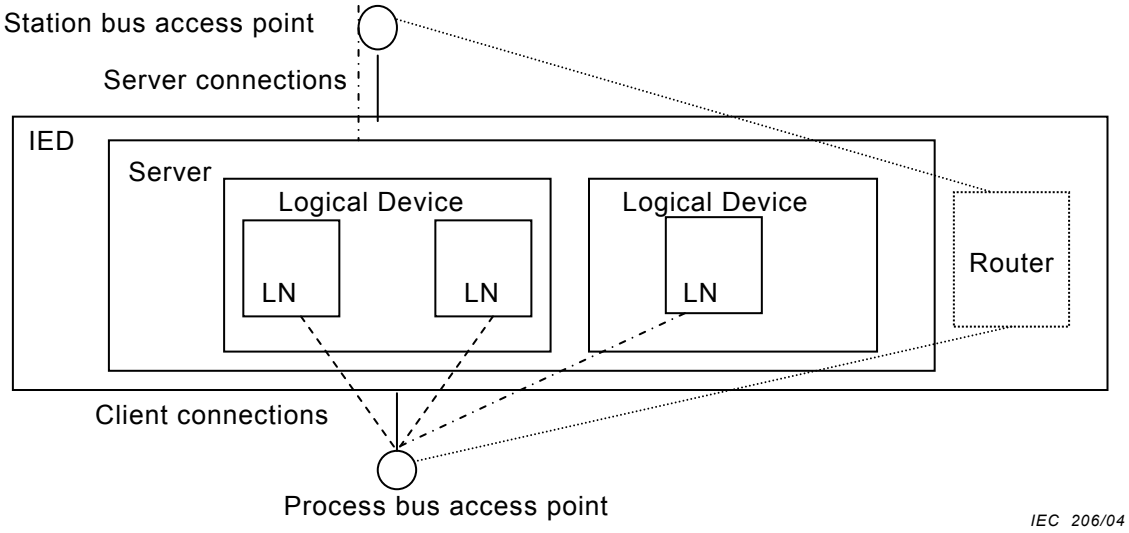


Figure 12 – IED structure and access points

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.

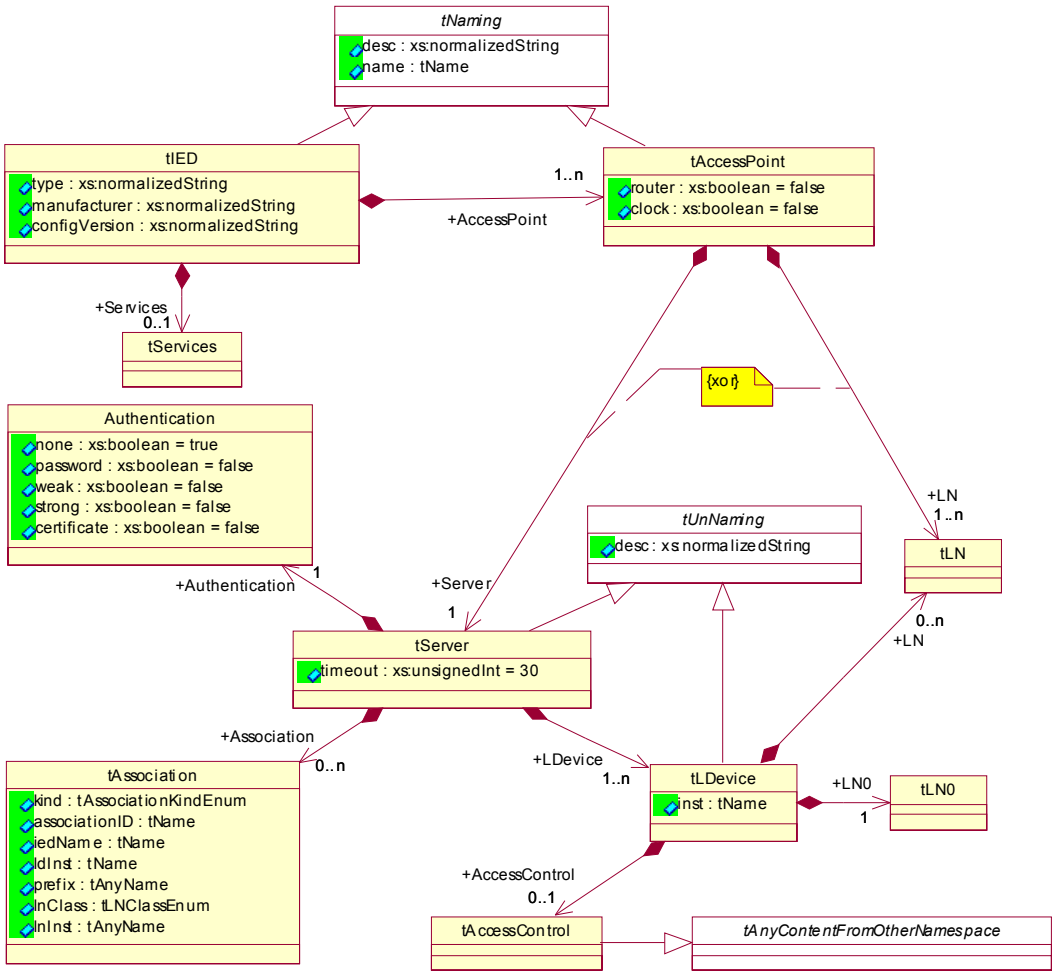
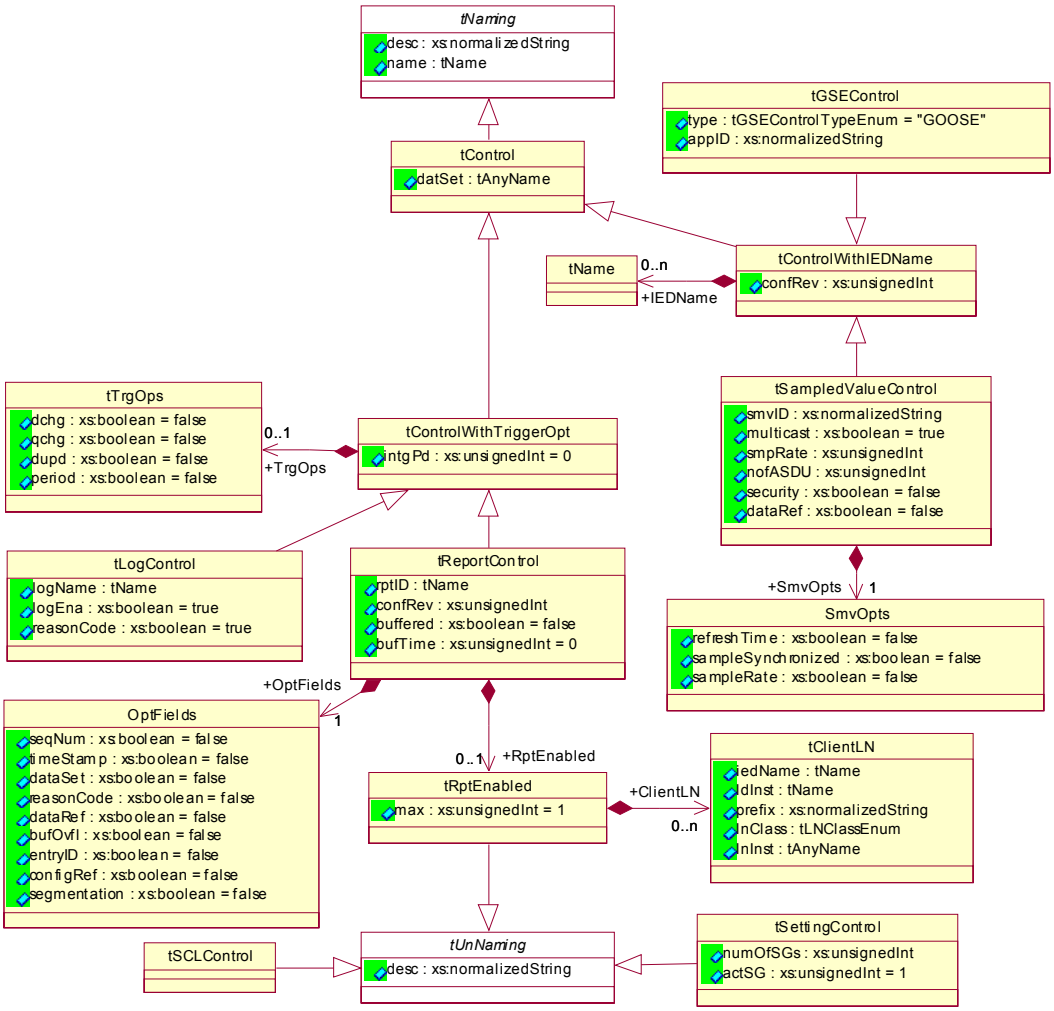


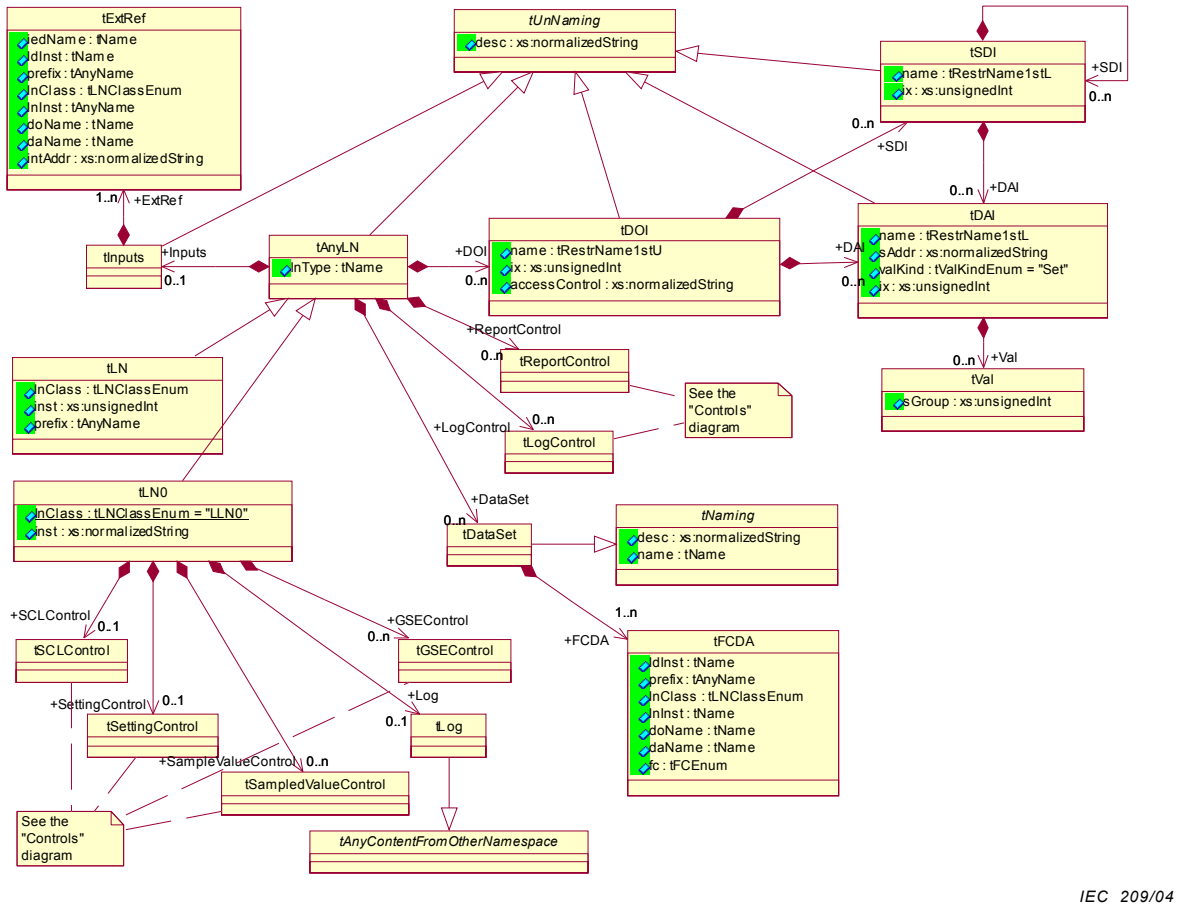
Figure 13 – UML description of IED related schema part – base

IEC 207/04



IEC 208/04

Figure 14 – UML description of IED related schema part for Control blocks



IEC 209/04

Figure 15 – UML description of IED related schema part – LN definition

9.3.2 The IED, Services and Access Point

The SCL syntax to describe an IED is as follows:

```
<xs:complexType name="tIED">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:sequence>
                <xs:element name="Services" type="tServices" minOccurs="0"/>
                <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
                    <xs:unique name="uniqueLNInAccessPoint">
                        <xs:selector xpath=".//scl:LN"/>
                        <xs:field xpath="@inst"/>
                        <xs:field xpath="@lnClass"/>
                        <xs:field xpath="@prefix"/>
                    </xs:unique>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="manufacturer" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="configVersion" type="xs:normalizedString" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes of the IED element are defined in Table 9.

Table 9 – Attributes of the IED element

Attribute name	Description
name	The identification of the IED. Within an ICD file describing a device type, the name shall be TEMPLATE. The IED name cannot be an empty string and shall be unique within an SCL file.
desc	The description text
type	The (manufacturer specific) IED product type
manufacturer	The manufacturer's name
configVersion	The basic configuration version of this IED configuration

The IED ConfigVersion above only identifies the IED basic configuration (its capabilities as defined/delivered by the manufacturer), and not its individual configuration after instantiation into a project. This is a parameter of the IED, or of its logical nodes. It shall be contained in an SCL file as an attribute value of the attribute *LLN0.NamPlt.configRev*. The IED contains a Service capability list, and access point definitions.

Restrictions

- The IED Name shall be unique within the IED section of the SCL file.
- The length of the IED Name shall be at least one.
- The IED name for an IED template shall be TEMPLATE.

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

- Within an *IED*, there cannot be two *AccessPoint* elements with same *name*.
- Within an *IED*, there cannot be two *LDevice* elements with the same *inst*. Moreover, the *inst* attribute of a *LDevice* acts as a key within the *IED*. The *logName* attribute of each *LogControl* (an indirect descendent of an *IED*), refers to one of such keys.

The Services element of the IED defines the available services.

```
<xs:complexType name="tServices">
    <xs:all>
        <xs:element name="DynAssociation" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="SettingGroups" minOccurs="0">
            <xs:complexType>
                <xs:all>
                    <xs:element name="SGEdit" type="tServiceYesNo" minOccurs="0"/>
                    <xs:element name="ConfSG" type="tServiceYesNo" minOccurs="0"/>
                </xs:all>
            </xs:complexType>
        </xs:element>
        <xs:element name="GetDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="GetDataObjectDefinition" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="DataObjectDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="GetDataSetValue" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="SetDataSetValue" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="DataSetDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfDataSet" type="tServiceWithMaxAndMaxAttributes" minOccurs="0"/>
        <xs:element name="DynDataSet" type="tServiceWithMaxAndMaxAttributes" minOccurs="0"/>
        <xs:element name="ReadWrite" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="TimerActivatedControl" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfReportControl" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="GetCBValues" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfLogControl" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="ReportSettings" type="tReportSettings" minOccurs="0"/>
        <xs:element name="LogSettings" type="tLogSettings" minOccurs="0"/>
```

```
            <xs:element name="GSESettings" type="tGSESettings" minOccurs="0"/>
            <xs:element name="SMVSettings" type="tSMVSettings" minOccurs="0"/>
            <xs:element name="GSEDir" type="tServiceYesNo" minOccurs="0"/>
            <xs:element name="GOOSE" type="tServiceWithMax" minOccurs="0"/>
            <xs:element name="GSSE" type="tServiceWithMax" minOccurs="0"/>
            <xs:element name="FileHandling" type="tServiceYesNo" minOccurs="0"/>
            <xs:element name="ConfLNs" type="tConfLNs" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
```

Service classes may appear in arbitrary order. If they do not appear, then the services are not available at the IED. If the same service name appears several times, this is of no meaning. For the meaning of the services, refer to IEC 61850-7-2.

The list of service capabilities and setting elements and attributes are defined in Table 10.

Table 10 – List of service capabilities and setting elements and attributes

Service capability	Description
DynAssociation	All services for dynamic building of associations. These are capabilities without attributes.
SettingGroups: SGEdit ConfSG	Setting group services belong to the setting group control block. If this control block is available, then the setting group service SelectActiveSG for activating a setting group is also available. The capability of online editing (IEC 61850-7-2 services SelectEditSG, ConfirmEditSGValues, SetSGValues) is decided with the SGEdit element. The capability to configure the (number of) setting groups by SCL can be also available (ConfSG). These are options without attributes.
GetDirectory	Service for reading the contents of a server, that is the LD and LN directories (all LDs, LNs and DATA of the LNs). This is an option without attributes. Includes the IEC 61850-7-2 services GetServerDirectory, GetLogicalDeviceDirectory, GetLogicalNodeDirectory.
GetDataObjectDefinition	Service to retrieve the complete list of all DA definitions of the referenced data that are visible and thus accessible to the requesting client by the referenced LN. It is a service without attributes. Refers to IEC 61850-7-2 service GetDataDefinition.
DataObjectDirectory	Service to get the DATA defined in a LN. It is a service without attributes. Refers to IEC 61850-7-2 service GetDataDirectory.
GetDataSetValue	Service to retrieve all values of data referenced by the members of the data set. It is a service without attributes. Refers to IEC 61850-7-2 service GetDataSetValues.
SetDataSetValue	Service to write all values of data referenced by the members of the data set. It is a service without attributes. Refers to IEC 61850-7-2 service SetDataSetValues.
DataSetDirectory	Service to retrieve FCD/FCDA of all members referenced in the data set. It is a service without attributes. Refers to IEC 61850-7-2 service GetDataSetDirectory.
ConfDataSet	If ConfDataSet is not specified, then the default value of its max attribute is equal to the number of preconfigured data sets, and they may be modified. If it is specified, it is possible to configure new data sets up to the defined max, or modify existing ones at configuration time via SCL. The attribute meaning is: max – the maximum number of data sets maxAttributes – the maximum number of attributes allowed in a data set (an FCDA can contain several attributes) modify – TRUE means that preconfigured data sets may be modified
DynDataSet	Services to dynamically create and delete data sets. The attribute meaning is: max – the maximum number of dynamically creatable data sets (including eventually predefined data sets) maxAttributes – the maximum number of attributes allowed in a data set (an FCDA can contain several attributes)
ReadWrite	Basic data read and write facility; includes the IEC 61850-7-2 services GetData, SetData, and the Operate service, if appropriate data exist. It is a capability without attributes.

TimerActivatedControl	This element specifies that timer activated control services are supported. All other control related services are specified directly at a DO with the ctlModel attribute. It is a service without attributes.
ConfReportControl	Capability of static (by configuration via SCL) creation of report control blocks. The attribute's meaning is: max – the maximum number of instantiable report control blocks
GetCBValues	Read values of control blocks. It is a service without attributes.
ConfLogControl	Capability of static (only while configuration via SCL) creation of log control blocks. The attribute's meaning is: max – maximum number of instantiable log control blocks.
ReportSettings	The report control block attributes for which online setting is possible with services SetURCBValuies respectively SetBRCBValues: The attribute's meaning is: cbName – control block name datSet – data set reference rptID – report identifier optFields – optional fields to include in report bufTime – buffer time trgOps – trigger options enable intgPd – integrity period
LogSettings	The log control block attributes for which online setting is possible with service SetLCBValues: The attribute's meaning is: cbName – control block name datSet – data set reference logEna – log enable trgOps – trigger options intgPd – integrity period
GSESettings	The GSE control block attributes for which online setting is possible with service SetGsCBValues respectively SetGoCBValues: The attribute's meaning is: cbName – control block name datSet – data set reference appID – application identifier dataLabel – value for the object reference if the corresponding element ist being sent (applies only to GSSE control blocks)
SMVSettings	The SMV control block attributes for which online setting is possible with service SetMSVCBValues repectively SetUSVCBValues: The attribute's meaning is: cbName – control block name datSet – data set reference svID – sample value identifier optFields – optional fields to include in sample value message smpRate – sample rate
ConfLNs	Describes what can be configured for LNs defined in an ICD file The attribute meanings are: fixPrefix – if false, prefixes can be set/changed fixLnInst – if false. LN instance numbers can be changed
GSEDir	GSE directory services according to IEC 61850-7-2. This capability has no attributes.

GOOSE	This element shows that the IED can be a GOOSE server and/or client according to IEC 61850-7-2. The attribute's meaning is: max = maximum number of GOOSE control blocks, which are configurable for publishing (max=0 means the device is only a GOOSE client).
GSSE	This element shows that the IED can be a binary data GSSE server and/or client according to IEC 61850-7-2. The attribute's meaning is: max – maximum number of GSSE control blocks, which are configurable.
FileHandling	All file handling services; without attributes.
NOTE Within an IED capability description, the maximum numbers specified above shall be a guaranteed maximum, i.e. this number of elements shall be possible to instantiate respective use under all circumstances, for example even if some dynamic memory allocation allows sometimes to have more elements (than maximum) of one type at the cost of another element type (always at least maximum).	

The Access point element of the IED defines the available communication access points.

```
<xs:complexType name="tAccessPoint">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:choice minOccurs="0">
                <xs:element name="Server" type="tServer">
                    <xs:unique name="uniqueAssociationInServer">
                        <xs:selector xpath="./scl:Association"/>
                        <xs:field xpath="@associationID"/>
                    </xs:unique>
                </xs:element>
                <xs:element ref="LN" maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute name="router" type="xs:boolean" use="optional" default="false">
            </xs:attribute>
            <xs:attribute name="clock" type="xs:boolean" use="optional" default="false">
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The Access point is decribed by one of the elements: *Server* or *LN* list.

The attributes of the Access point element are defined in Table 11.

Table 11 – Attributes of the Access point element

Attribute name	Description
name	Reference identifying this access point within the IED
desc	The description text
router	The presence and setting to true defines this IED to have a router function. By default, its value is false (no router function).
clock	The presence and setting to true defines this IED to be a master clock at this bus. By default, its value is false (no master clock).

The name attribute of the access point together with the name of the IED gives a unique reference for the access point within the SA system.

If neither a router, nor a clock, nor a server, nor a LN list is specified, the access point may only be used by client LNs in the same IED to access the bus to which it is connected. This is typical for a process bus access point of a bay level device, where the LNs offer their data via a server to the station bus only.

Project-specific access point attributes, such as the address within a communication system, are contained in the SCL Communication section.

Restrictions

- The name of the access point shall be unique within the IED.
- The name shall not be empty.

Note that

- an IED can be purely a router or a clock, if it does not contain any other element (especially a server),
- an additional router or clock function may exist on a server access point,
- in the most common case, the IED contains only the server,
- if the IED contains only a LN list, these are clients only and the IED can not be supervised, because no server offers the appropriate data. An additional router or clock function is possible.

9.3.3 The IED server

A communication server of the IED will described as follows:

```
<xs:complexType name="tServer">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Authentication">
                    <xs:complexType>
                        <xs:attributeGroup ref="agAuthentication"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="LDevice" type="tLDevice" maxOccurs="unbounded"/>
                <xs:element name="Association" type="tAssociation" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="timeout" type="xs:unsignedInt" use="optional" default="30"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The IED server contains the elements *Authentication, LDevice* and *Association*. The attributes are defined as shown in Table 12.

Table 12 – Attributes of the IED server element

Attribute name	Description
timeout	Time out in seconds: if a started transaction (for example selection of a setting group) is not completed within this time, it is cancelled and reset
desc	A descriptive text

The server is identified within the system by the access point to which it belongs. Its identification in the communication system (address) is contained in the SCL communication section (see 9.4).

The mandatory Authentication element defines in the case of a device description the authentication possibilities, in case of a device instantiated in a plant the method(s) to be used for authentication. If the element is missing, the default value is *none* (that is no authentication, meaning that the attribute *none* has the value *true*). The exact meaning of the other methods, especially weak and strong, is defined in the stack mappings (SCSMs).

```
<xs:attributeGroup name="agAuthentication">
    <xs:attribute name="none" type="xs:boolean" use="optional" default="true"/>
    <xs:attribute name="password" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="weak" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="strong" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="certificate" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
```

The attributes of the Authentication element are defined in Table 13.

Table 13 – Attributes of the Authentication element

Attribute name	Description
none	No authentication
password	Is defined in the stack mappings (SCSMs)
weak	
strong	
certificate	

9.3.4 The logical device

The *LDevice* element defines a logical device of the IED reachable via the access point. It shall contain at least one LN and the LN0, and may contain preconfigured report, GSE and SMV definitions.

```
<xs:complexType name="tLDevice">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element ref="LN0"/>
                <xs:element ref="LN" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="AccessControl" type="tAccessControl" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="inst" type="tName" use="required">
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes of the LDevice element are defined in Table 14.

Table 14 – Attributes of the LDevice element

Attribute name	Description
inst	Identification of the LDevice within the IED. The full LD name according to IEC 61850-7-x contains an additional part before this *inst* value (see also 8.4). Its value cannot be the empty string
desc	The description text

Restrictions

- The LD *inst* shall be unique within the IED.
- The LD name built from *inst* and other parts as described in 8.4 shall be unique within each SCL file.
- The length of the attribute *inst* shall be at least one.

9.3.5 LN0 and other logical nodes

```
<xs:complexType name="tLN0">
    <xs:complexContent>
        <xs:extension base="tAnyLN">
            <xs:sequence>
                <xs:element name="GSEControl" type="tGSEControl" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="SampledValueControl" type="tSampledValueControl" minOccurs="0"
                 maxOccurs="unbounded"/>
                <xs:element name="SettingControl" type="tSettingControl" minOccurs="0"/>
                <xs:element name="SCLControl" type="tSCLControl" minOccurs="0"/>
                <xs:element name="Log" type="tLog" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="lnClass" type="tLNClassEnum" use="required" fixed="LLN0"/>
            <xs:attribute name="inst" type="xs:normalizedString" use="required" fixed=""/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The LN0 contains the following elements: *GSEControl* (see 9.3.10), *SampledValueControl* (see 9.3.11), *SettingControl* (see 9.3.12), *SCLControl* and *Log*. Furthermore, it inherits *ReportControl* and *LogControl* from the base type tAnyLN, as well as the *DOI* and *Inputs* element.

The attributes of the LN0 element are defined in Table 15.

Table 15 – Attributes of the LN0 element

Attribute name	Description
lnClass	The LN class according to IEC 61850-7-x and also defined in tAnyLN is here fixed to LLN0, i.e. no other value is allowed
lnType	The instantiable type definition of this logical node, reference to a LNodeType definition
inst	The LN instance number identifying this LN. For LLN0 it is fixed to be the empty string (no other value is allowed)
desc	The description text

Restrictions

- The LN0 LN class is always LLN0, so no *inst* attribute is needed (default is the empty string). For the referencing of links to LN0, *lnInst* shall be the empty string, and *lnClass* shall be LLN0.

The Logical Node (type *tLN*) is described as follows:

```
<xs:complexType name="tLN">
    <xs:complexContent>
      <xs:extension base="tAnyLN">
        <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
        <xs:attribute name="inst" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="prefix" type="tAnyName" use="optional" default=""/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

tAnyLN, the super-type of both *tLN0* and *tLN*, is defined as follows:

```
<xs:complexType name="tAnyLN" abstract="true">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="DataSet" type="tDataSet" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="ReportControl" type="tReportControl" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="LogControl" type="tLogControl" minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xs:element name="DOI" type="tDOI" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Inputs" type="tInputs" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="lnType" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The LN contains the following elements: DataSet (see 9.3.7), ReportControl (see 9.3.8), LogControl (see 9.3.9), DOI (see 9.3.6) and Inputs (see 9.3.13).

The attributes of the LN are defined as shown in Table 16.

Table 16 – Attributes of the LN element

Attribute name	Description
desc	The description text for the logical node
lnType	The instantiable type definition of this logical node, reference to a LNodeType definition
lnClass	The LN class according to IEC 61850-7-x
inst	The LN instance number identifying this LN – an unsigned integer
prefix	The LN prefix part

The optional DOI elements in a LN definition can be used to define special instance related values for DATA and their attributes by using SDI elements for DATA or attribute structure parts (if needed) and DAI elements per final attribute (see DOI definition in 9.3.6). The DATA and attributes referenced here shall however already be defined within the LNodeType definition of the LN, referenced with the LNType attribute of the LN. The DOI elements at this place for this instance shall NOT define new DOs or new attributes, which are not contained in the LNodeType. For example, the pulse length configuration parameter of a DPC CDC, specified with 100 ms in the LNodeType, is overwritten here with a value of 300 ms for this special DO.

Restrictions

- The LN Name consisting of *prefix, lnClass* and *inst* shall be unique within the scope of the logical device, if a server is defined, or else within the scope of the IED.

9.3.6 DATA (DOI) definition

```
<xs:complexType name="tDOI">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="SDI" type="tSDI"/>
                    <xs:element name="DAI" type="tDAI"/>
                </xs:choice>
                <xs:attribute name="name" type="tRestrName1stU" use="required"/>
                <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
                <xs:attribute name="accessControl" type="xs:normalizedString" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

The DOI is decribed by one of the element: *SDI* or *DAI*.

The attributes of the DOI are defined as shown in Table 17.

Table 17 – Attributes of the DOI element

Attribute name	Description
desc	The description text for the data
name	A standardised DO name for example from IEC 61850-7-4
ix	Index of a data element in case of an array type
accessControl	Access control definition for this data. The empty string (default) means that the higher-level access control definition applies. Possible values are SCSM dependent.

The DAI attribute within the DOI defines the attributes and the related values to be set. Again, all attributes shall also be contained in the LNodeType definition of this LN. Only those are repeated here, where some aditional (attribute or element) values shall be set or individually overwritten.

```
<xs:complexType name="tDAI"
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Val" type="tVal" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" type="tRestrName1stL" use="required"/>
            <xs:attribute name="sAddr" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="valKind" type="tValKindEnum" use="optional" default="Set"/>
            <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The DAI contains the elements *Val* (see 9.5.4).

The DAI allows the description of instance values for an IED. This can be used at the engineering stage by other IEDs/LNs which need to know configuration related values, for example if they have no services to read the values, or if the IED does not support their reading. Alternatively it can be used by the IED itself to set these values, either to offer them via the communication protocol, or at least consider them in its internal functions.

The attributes of the DAI are defined as shown in Table 18.

Table 18 – Attributes of the DAI element

Attribute name	Description
desc	The description text for the DAI element
name	The name of the Data attribute whose value is given
sAddr	Short address of this Data attribute
valKind	The meaning of the value from the engineering phases, if any name is given
ix	Index of the DAI element in case of an array type

The DAI element contains a subset of the DA attributes, and shall be used within an IED DOI specification if some instance specific attribute values are set or typical attribute values overwritten.

The subset of data or data attributes are described as follows:

```
<xs:complexType name="tSDI">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="SDI" type="tSDI"/>
```

```
            <xs:element name="DAI" type="tDAI"/>
        </xs:choice>
        <xs:attribute name="name" type="tRestrName1stL" use="required"/>
        <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The SDI element stands for a substructure name part, either from a DO (corresponding to SDO in LNodeType) or a DA substructure name, except the final (leaf) attribute name. The SDI element contains either the elements *SDI* for a further structure name part,or *DAI* for the final attribute element with the value(s).

The attributes of the SDI element are defined as shown in Table 19.

Table 19 – Attributes of the SDI element

Attribute name	Description
desc	A description text for the SDI part
name	Name of the SDI (structure part)
ix	Index of the SDI element in case of an array type

Restrictions

The name shall begin with a lower-case letter.

Example:

The following example describes the value of a structured DO as DOI

```
<DOI name="Volts">
    <SDI name="sVC">
        <DAI name="offset"><Val>0</Val></DAI>
        <DAI name="scaleFactor"><Val>200</Val></DAI>
    </SDI>
</DOI>
```

9.3.7 Data set definition

```
<xs:complexType name="tDataSet">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="FCDA" type="tFCDA" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The DataSet contains a sequence of FCDA elements.

The data set definition of the LN has the following attributes:

Table 20 – Attributes of the DataSet element

Attribute name	Description
name	A name identifying this data set in the LN where it is defined
desc	The description text for the data set

```
<xs:complexType name="tFCDA">
        <xs:attribute name="ldInst" type="tName" use="optional"/>
        <xs:attribute name="prefix" type="tAnyName" use="optional"/>
        <xs:attribute name="lnClass" type="tLNClassEnum" use="optional"/>
        <xs:attribute name="lnInst" type="tName" use="optional"/>
        <xs:attribute name="doName" type="tName" use="optional"/>
        <xs:attribute name="daName" type="tName" use="optional"/>
        <xs:attribute name="fc" type="tFCEnum" use="required"/>
</xs:complexType>
```

The FCDA element defines the name of a functionally constraint data or functionally constraint data attribute according to IEC 61850-7-2 of this IED to be contained in the data set. The order of the FCDA elements in the data set defines the order of data values within the communication messages, if no other rules or SCSM conventions apply. The element has the following attributes:

Table 21 – Attributes of the FCDA element

Attribute name	Description
ldInst	The LD where the DO resides
prefix	Prefix identifying together with *lnInst* and *lnClass* the LN where the DO resides
lnClass	LN class of the LN where the DO resides; shall always be specified except for GSSE DataLabel empty string
lnInst	Instance number of the LN where the DO resides; shall be specified except for LLN0
doName	A name identifying the DO (within the LN). A name standardized in IEC 61850-7-4. If *doName* is empty, then *fc* can contain a value, selecting the attribute category of all DOs of the defined LN. For elements or parts of structured DATA types, all name parts are contained, separated by dots (.)
daName	The attribute name – if empty, all attributes with functional characteristic given by *fc* are selected. For elements or parts of structured data types, all name parts are contained, separated by dots (.)
fc	All attributes of this functional constraint are selected. Possible constraint values see IEC 61850-7-2 or the *fc* definition in 9.5

If *daName* and *fc* both contain a non empty value, then the *fc* value must be valid for the attribute (i.e. defined identically at the appropriate LNodeType definition), otherwise the SCL file processing shall be stopped with an error message. If all attributes of the FCDA (except *fc*) are missing or empty, then this corresponds to an empty string in a GSSE DataLabel definition (*fc* value should be ST) – in all other data sets, this is not allowed.

All control blocks, which reference a data set, shall be contained in the same LN as the data set definition. Therefore, the data set reference within all control blocks only contains the LN relative data set name (Name attribute at DataSet element), and not its full name (which also contains the LD name and LN name according to IEC 61850-7-2).

If the order of data within a message based on this data set definition is given, then the FCDA order in the data set shall be followed. If a set of attributes is specified for example via *fc*, then the order of data is specified by the DATA and attribute order in the correspondfing LNs *LnodeType*.

9.3.8 Report control block

A report control block definition of the LN is as follows:

```
<xs:complexType name="tReportControl">
        <xs:complexContent>
            <xs:extension base="tControlWithTriggerOpt">
                <xs:sequence>
                    <xs:element name="OptFields">
                        <xs:complexType>
```

```
                        <xs:attributeGroup ref="agOptFields"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="RptEnabled" type="tRptEnabled" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="rptID" type="tName" use="required"/>
            <xs:attribute name="confRev" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="buffered" type="xs:boolean" use="optional" default="false"/>
            <xs:attribute name="bufTime" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="tControlWithTriggerOpt" abstract="true">
    <xs:complexContent>
        <xs:extension base="tControl">
            <xs:sequence>
                <xs:element name="TrgOps" type="tTrgOps" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="intgPd" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The report control block (RCB) contains the elements: *TrgOps*, *OptFields* and *RptEnabled*.

The attributes given in Table 22 are used.

Table 22 – Attributes of the report control block element

Attribute name	Description
name	Name of the report control block. This name is relative to the LN hosting the RCB, and shall be unique within the LN
desc	The description text
datSet	The name of the data set to be sent by the report control block; datSet can only be empty within an ICD-File.
intgPd	Integrity period in milliseconds – see IEC 61850-7-2. Only relevant if trigger option *period* is set to true
rptID	Identifier for the report control block
confRev	The configuration revision number of this report control block
buffered	Specifies if reports are buffered or not – see IEC 61850-7-2
bufTime	Buffer time – see IEC 61850-7-2

The attributes of element *TrgOps* are defined as follows:

```
<xs:complexType name="tTrgOps">
    <xs:attribute name="dchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="qchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dupd" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="period" type="xs:boolean" use="optional" default="false"/>
</xs:complexType>
```

If an attribute is not given, its value (the corresponding trigger option) is false, meaning that the trigger option shall not be used.

The element OptFields is defined as follows:

```
<xs:element name="OptFields">
    <xs:complexType>
        <xs:attributeGroup ref="agOptFields"/>
    </xs:complexType>
</xs:element>
```

```
<xs:attributeGroup name="agOptFields">
        <xs:attribute name="seqNum" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="timeStamp" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dataSet" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="reasonCode" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dataRef" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="bufOvfl" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="entryID" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="configRef" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="segmentation" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
```

Setting one of the attributes to true means that the corresponding data shall be included into the report (see IEC 61850-7-2).

The element RptEnabled is defined as follows:

```
<xs:complexType name="tRptEnabled">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="ClientLN" type="tClientLN" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="max" type="xs:unsignedInt" use="optional" default="1"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

The RptEnabled element contains the list of client LNs for which this report shall be enabled (for example at IED startup on pre-established associations).

The attributes given in Table 23 are used.

Table 23 – Attributes of the RptEnabled element

Attribute name	Description
desc	The description text
max	Defines the maximum number of report control blocks of this type, which are instantiated at configuration time in the LN (and then used online)

According to IEC 61850-7-2, a report control block is dedicated to at most one client at a time. This means that if *Max* > 1 is given for *RptEnabled*, more than one report control block (RCB) of this type has to be instantiated in the IED. Observe that for all buffered control blocks, a ClientLN shall be preconfigured, i.e. *Max* is identical to the number of ClientLNs given. If ClientLNs are preconfigured for unbuffered RCBs, then the *Resv* (URCB Reservation is described in IEC 61850-7-2) attribute of the RCB shall be set to true additionally to the *RptEna* attribute (Report Enable is described in IEC 61850-7-2) in the IED. The URCName or BRCName of the control block according to IEC 61850-7-2 is built from the RCName attribute above followed by a two digit number between 01 and *Max*. If ClientLNs are defined, the index (position) of the ClientLN in the list contained in the RptEnabled element is used as this number for this client (the first has index 1). This means that a report control block definition in SCL has to be considered as a type, and not as an instance, which might have 99 instances to 99 clients.

The ClientLN element defines the name of a LN in the system, which is a client to this report CB type.

```
<xs:complexType name="tClientLN">
        <xs:attributeGroup ref="agLNRef"/>
</xs:complexType>
```

```
<xs:attributeGroup name="agLNRef">
        <xs:attributeGroup ref="agLDRef"/>
        <xs:attribute name="prefix" type="xs:normalizedString" use="optional"/>
        <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
        <xs:attribute name="lnInst" type="xs:normalizedString" use="required"/>
</xs:attributeGroup>
```

The attributes given in Table 24 are used.

Table 24 – Attributes of the ClientLN element

Attribute name	Description
iedName	The name of the IED where the LN resides
ldInst	The instance identification of the LD where the LN resides
prefix	The LN prefix
lnClass	The LN class according to IEC 61850-7-4
lnInst	The instance id of this LN instance of below LN class in the IED

Restrictions

The name of the report control block shall be unique within the LN.

Note that to identify a LN within the system, the IED based designation is used here, even if the communication name generation is based on the substation functional structure. It is recommended that a tool assures that the defined client is really accessible across the defined communication system.

For pre-established associations, the AssociationId corresponding to the referenced LN can be found in the association definition section of this IED.

Example:

```
    <ReportControl name="PosReport" rptID="E1Q1Switches" datSet="Positions" confRev="0">
        <TrgOps dchg="true" qchg="true"/>
        <OptFields/>
        <RptEnabled max="5">
            <ClientLN iedName="A1KA1" ldInst="LD1" lnInst="1" lnClass="IHMI"/>
        </RptEnabled>
    </ReportControl>
```

The RptEnabled part defines that the Report control block type is valid for 5 (unbuffered) RCBs with names PosReport01, PosReport02, up to PosReport05. The first one, PosReport01, is already reserved for the client A1KA1LD1/IHMI1. All reports are triggered with dchg and qchg, and the buffer time is 0. No OptFields are defined, i.e. only the mandatory information is included in the report.

9.3.9 Log control block

A log control block is defined with the following element:

```
<xs:complexType name="tLogControl">
        <xs:complexContent>
            <xs:extension base="tControlWithTriggerOpt">
                <xs:attribute name="logName" type="tName" use="required"/>
                <xs:attribute name="logEna" type="xs:boolean" use="optional" default="true"/>
                <xs:attribute name="reasonCode" type="xs:boolean" use="optional" default="true"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="tControlWithTriggerOpt" abstract="true">
    <xs:complexContent>
        <xs:extension base="tControl">
            <xs:sequence>
                <xs:element name="TrgOps" type="tTrgOps" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="intgPd" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The meaning of the attributes is mostly identical to the appropriate control block attributes defined in IEC 61850-7-2. For those where it is completely identical the same attribute name is used.

The attributes of the log control block element are defined in Table 25.

Table 25 – Attributes of the log control block element

Attribute name	Description
name	the name of the log control block
desc	a description text
datSet	the name of the data set whose values shall be logged; datSet can only be empty within an ICD-File.
intgPd	integrity scan period in milliseconds – see IEC 61850-7-2.
logName	Reference to the LD, which is the owner of the log
logEna	TRUE enables immediate logging; FALSE prohibits logging until enabled online
reasonCode	reason code – see IEC 61850-7-2

Restrictions

The name of the log control block shall be unique within the LN.

The following extract of an SCL file shows a log control block example, which logs data from the data set Positions into the log of the logical device C1, triggered by either data change or quality change.

```
<LogControl name="Log" datSet="Positions" logName="C1">
    <TrgOps dchg="true" qchg="true"/>
</LogControl>
```

9.3.10 GSE control block

The following GSE control element is only allowed in the logical node LLN0.

```
<xs:complexType name="tGSEControl">
        <xs:complexContent>
            <xs:extension base="tControlWithIEDName">
                <xs:attribute name="type" type="tGSEControlTypeEnum" use="optional" default="GOOSE"/>
                <xs:attribute name="appID" type="xs:normalizedString" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

<xs:complexType name="tControlWithIEDName">
        <xs:complexContent>
            <xs:extension base="tControl">
                <xs:sequence>
                    <xs:element name="IEDName" type="tName" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="confRev" type="xs:unsignedInt" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

The GSE control block may optionally contain IED names for those IEDs which have to subscribe the GSE data.

The attributes given in Table 26 are used.

Table 26 – Attributes of the GSE control block element

Attribute name	Description
name	The name identifying this GOOSE control block.
desc	A description text.
datSet	The name of the data set to be sent by the GSE control block. For type=GSSE, the FCDA definitions in this data set shall be interpreted as DataLabels according to IEC 61850-7-2. The attribute datSet can only be empty within an ICD-File.
confRev	The configuration revision number of this control block.
type	If the *type* is *GSSE*, then only single indication and double indication data types are allowed for the data items referenced in the data set, otherwise all data types are allowed. Note that on stack level, each type might be mapped differently to message formats. The default type value is GOOSE.
appID	A system wide unique identification of the application to which the GOOSE message belongs.

Restrictions

- The GSE control block name shall be unique within the LLN0, i.e. the logical device.
- Different applications within the station shall have unique appId values. It is up to the project/system engineer to decide what an application is.

The following SCL extract shows an example of a GOOSE control block definition:

```
<GSEControl name="ItlPositions" datSet="Positions" appID="Itl"/>
```

Its relative name within this LN0 is *ItlPositions*, its message contents is defined by the data set *Positions*, and it shall be used for the *Itl* application.

9.3.11 Sampled value control block

The following sampled value control block element is only allowed in the logical node LLN0.

```
<xs:complexType name="tSampledValueControl">
    <xs:complexContent>
        <xs:extension base="tControlWithIEDName">
            <xs:sequence>
                <xs:element name="SmvOpts">
                    <xs:complexType>
                        <xs:attributeGroup ref="agSmvOpts"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="smvID" type="xs:normalizedString" use="required"/>
            <xs:attribute name="multicast" type="xs:boolean" default="true"/>
            <xs:attribute name="smpRate" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="nofASDU" type="xs:unsignedInt" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The sampled value control block contains the element SmvOpts, and as extension of the schema type *tControlWithIEDName* optional several IED names of IEDs which shall receive the messages.

The attributes given in Table 27 are used.

Table 27 – Attributes of the sampled value control block element

Attribute name	Description
name	A name identifying this SMV control block
desc	The description text
datSet	The name of the data set whose values shall be sent; datSet can only be empty within an ICD-File
confRev	The configuration revision number of this control block
smvID	Multicast CB: the MsvID for the sampled value definition as defined in IEC 61850-7-2 Unicast CB: the UsvID as defined in IEC 61850-7-2
multicast	*false* indicates Unicast SMV services only meaning that smvID = UsvID
smpRate	Sample rate as defined in IEC 61850-7-2
nofASDU	Number of ASDU (Application service data unit) – see IEC 61850-9-2

If Multicast is FALSE, i.e. this is a Unicast control block, the definition has to be considered as a type. In this case, then

- for each IED name of a destination IED contained in the definition a control block instance shall be created,
- The *UsvCBName* defined in IEC 61850-7-2 shall be set to this IED name concatenated with the *name* above (which might then be empty),
- the *Resv* attribute of the CB shall be initialized to TRUE.

If Multicast is TRUE, then *name* corresponds directly to MsvCBName.

The following attributes can be set:

```
<xs:attributeGroup name="agSmvOpts">
    <xs:attribute name="refreshTime" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="sampleSynchronized" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="sampleRate" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="security" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dataRef" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
```

The attributes of the Smv Options element are defined in Table 28.

Table 28 – Attributes of the Smv Options element

Attribute name	Description
refreshTime	The meaning of the options is described in IEC 61850-7-2. If any of the attributes is set to true, the appropriate values shall be included into the SMV telegram
sampleSynchronized	
sampleRate	
security	See IEC 61850-9-2 for description
dataRef	If true, then the data set reference is included in the SV message

Restrictions

- The SV control block name shall be unique within the LLN0, i.e. within the LDevice.

The following SCL extract show the definition of an SV control block, which refers to data set smv. This data set defines the data contents of the SV message:

```
<SampledValueControl name="Volt" datSet="smv" smvID="11" smpRate="4800" nofASDU="5" multicast="true">
    <SmvOpts sampleRate="true" refreshTime="true" sampleSynchronized="true"/>
</SampledValueControl>
```

9.3.12 Setting control block

The following defines the definition for a setting control block (SGC). Note that the SGC name, i.e. its name part within the LN0, is SGCB according to IEC 61850-7-2. Therefore, only one SGC is allowed per LN0.

```
<xs:complexType name="tSettingControl">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:attribute name="numOfSGs" type="xs:unsignedInt" use="required"/>
                <xs:attribute name="actSG" type="xs:unsignedInt" use="optional" default="1"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

The attributes are identical to those of the setting group control block in IEC 61850-7-2.

The attributes of the setting control block element are defined in Table 29.

Table 29 – Attributes of the setting control block element

Attribute name	Description
desc	The description text
numOfSGs	The number of setting groups available
actSG	The number of the setting group to be activated when loading the configuration. The default value is 1

9.3.13 Binding to external signals

The Inputs section defines all external signals which are needed by the LN application to fulfill its function. The section also allows the binding of the signal to an internal address IntAdr.

```
<xs:complexType name="tInputs">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="ExtRef" type="tExtRef" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

Each ExtRef element references one external item, either at DO or at DA level. If IntAdr is needed, it has to be used appropriately to this level. This means that for a DO level usage it might contain a mapping of several attributes.

```
<xs:complexType name="tExtRef">
        <xs:attributeGroup ref="agDORef"/>
        <xs:attribute name="daName" type="tName" use="optional"/>
        <xs:attribute name="intAddr" type="xs:normalizedString" use="optional"/>
</xs:complexType>
```

The attributes shown in Table 30 are used.

Table 30 – Attributes of the Input/ExtRef element

Attribute name	Description
iedName	The name of the IED from where the input comes
ldInst	The LD instance name from where the input comes
prefix	The LN prefix
lnClass	The LN class according to IEC 61850-7-x
lnInst	The instance id of this LN instance of below LN class in the IED
doName	A name identifying the DO (within the LN).In case of structured DO, the name parts are concatenated by dots (.)
daName	The attribute designating the input. The IED tool should use an empty value if it has some default binding (IntAdr) for all process input attributes of a DO (*fc* = ST or MX), especially for t and q. If the attribute belongs to a data type structure, then the structure name parts shall be separated by dots (.)
intAddr	The internal address to which the input is bound. Only the IED tool of the concerned IED shall use the value. All other tools shall preserve it

An empty *daName* value means all the operational value attribute(s) of the DO, i.e. stVal, mag, etc. In this case, *intAdr* can also specify the addresses of all operational attributes in some IED tool specific way.

If the same input data can be received by the IED by different communication services (for example by report and by GSE), it is up to the IED or its tool implementation to decide which one shall be taken.

9.3.14 Associations

```
<xs:complexType name="tAccessControl" mixed="true">
        <xs:complexContent mixed="true">
            <xs:extension base="tAnyContentFromOtherNamespace"/>
        </xs:complexContent>
</xs:complexType>
```

An access control definition. Meaning and eventual refinement of the definition are stack (SCSM) specific issues.

It is recommended that all authorization and access control is done by private implementation within the interface LNs. In this case, no access control definitions are necessary within SCL.

Each association definition defines one pre-configured association between this server and a client logical node. Two kinds of pre-configuration are possible. *Predefined* means that this association is defined, but not yet opened, the client has to open it. *Pre-established* means that the association is defined and considered to be open directly after IED start up.

```
<xs:complexType name="tAssociation">
        <xs:attribute name="kind" type="tAssociationKindEnum" use="required"/>
        <xs:attribute name="associationID" type="tName" use="optional" />
        <xs:attributeGroup ref="agLNRef"/>
</xs:complexType>
```

The attributes shown in Table 31 are used.

Table 31 – Attributes of the Association element

Attribute name	Description
kind	The kind of pre-configured association, pre-established or predefined
associationID	The identification of a pre-configured association (otherwise empty)
iedName	The reference identifying the IED on which the client resides
ldInst	The reference to the client logical device
lnClass	The class of the client LN
prefix	The LN prefix
lnInst	The instance number of the client LN

An empty association Id as given by the default value means that the association Id is not yet defined. For a completed SCL file and a pre-established association, the association Id shall be set, so that the client LNs and the server can verify it correctly. The same client may use the same association to different LNs on the same server. Uniqueness requirements as well as value range of the association Id (for example a 32 bit integer, unique at the server, or at server IED and client Id, or system wide) are set up in the SCSMs.

Restrictions

- The Association ID shall be unique within the Server.
- The length of the Association ID shall be at least one.

9.4 Communication system description

9.4.1 General

This Clause describes the direct communication connection possibilities between logical nodes by means of logical buses (SubNetworks) and IED access points. The IED sections already describe which LDs and LNs are reachable across a certain access point. The communication section now describes which IED access points are connected to a common subnetwork. This is done in a way that reflects the hierarchical name structure within the IED, which is based on IED relative names for access points, LDs and LNs.

The UML diagram shown in Figure 16 gives an overview of the Communication section.

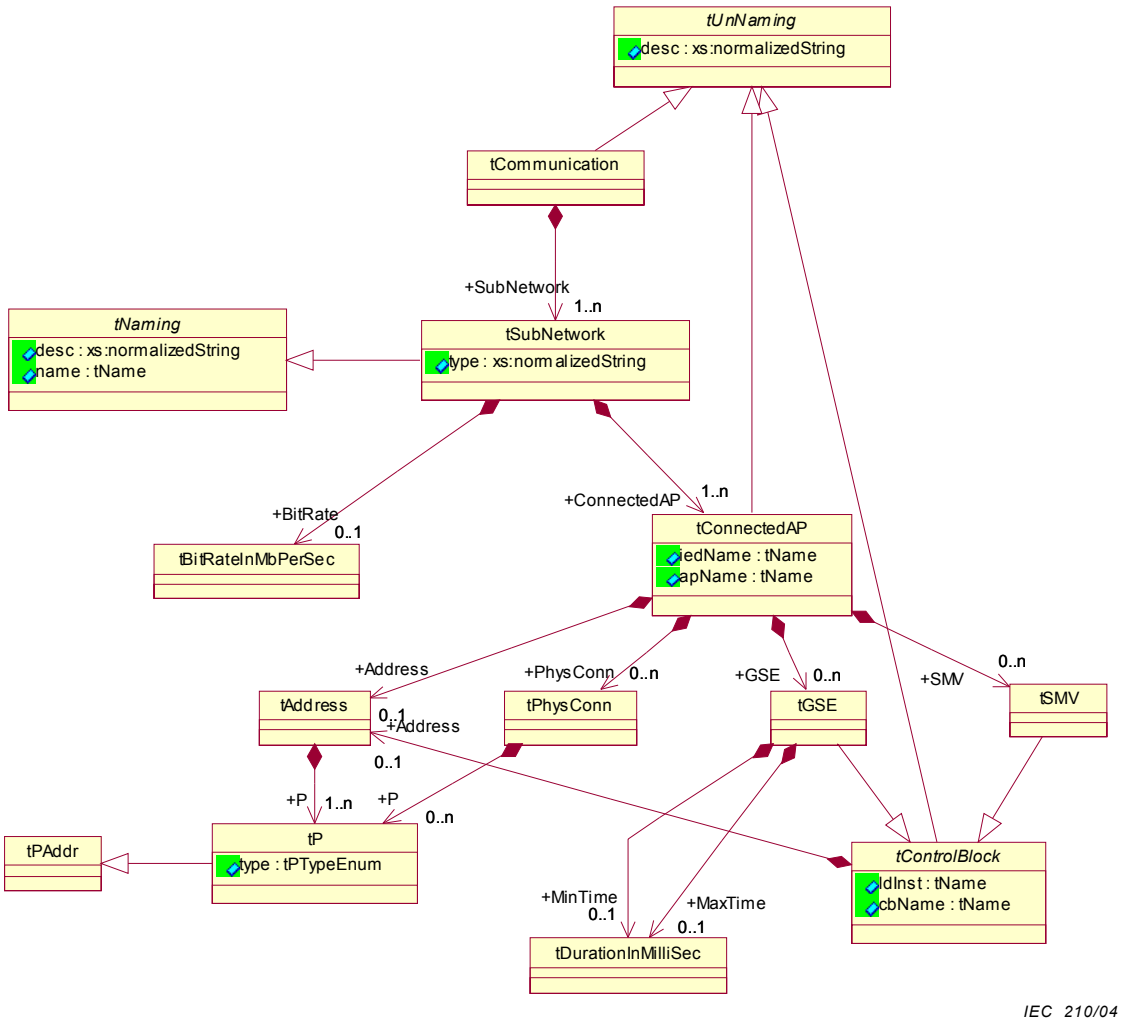


Figure 16 – UML diagram overview of the Communication section

Here follows the formal XML schema definition.

```
<xs:element name="Communication" type="tCommunication">
    <xs:unique name="uniqueSubNetwork">
        <xs:selector xpath="./scl:SubNetwork"/>
        <xs:field xpath="@name"/>
    </xs:unique>
</xs:element>

<xs:complexType name="tCommunication">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="SubNetwork" type="tSubNetwork" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The Communication section might optionally contain Text and Private sections (derivation from tUnNaming). The names of the SubNetworks shall be unique.

9.4.2 Subnetwork definition

A SubNetwork definition contains all access points which can (logically) communicate with the SubNetwork protocol and without the intervening router. Observe that a subnetwork defines a logical connection with a certain protocol. Different subnetworks with different protocols might run on the same physical communication network.

```
<xs:complexType name="tSubNetwork">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:sequence>
                <xs:element name="BitRate" type="tBitRateInMbPerSec" minOccurs="0"/>
                <xs:element name="ConnectedAP" type="tConnectedAP" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="type" type="xs:normalizedString" use="optional">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The bus protocol types are defined in IEC 61850 Part 8 and
9</xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes of a Subnetwork are defined as shown in Table 32.

Table 32 – Attributes of the Subnetwork element

Attribute	Description
name	A name identifying this bus; unique within this SCL file
desc	Some descriptive text to this SubNetwork
type	The SubNetwork protocol type; protocol types are defined by the SCSMs. In the examples, 8-MMS is used for the protocol defined in IEC 61850-8-1.

Protocol types are defined in the stack mappings (SCSM), IEC 61850-8-x and IEC 61850-9-x for this standard series. Those of IEC 61850-8-x start with “8-“, those of IEC 61850-9-x with “9-“ (except if they are identical). The protocol of IEC 61850-8-1 is for example 8-MMS, and IEC 61850-9-2 uses the same protocol.

The Subnetwork contains an optional BitRate element defining the bit rate in Mbit/s, and a list of IED access points by which these IEDs are connected to a SubNetwork with access points. It inherits Private and Text elements from tUnNaming.

```
<xs:complexType name="tConnectedAP">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Address" type="tAddress" minOccurs="0"/>
                <xs:element name="GSE" type="tGSE" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="SMV" type="tSMV" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="PhysConn" type="tPhysConn" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="iedName" type="tName" use="required"/>
            <xs:attribute name="apName" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The ConnectedAP is the IED access point connected to this SubNetwork.

It has the attributes shown in Table 33.

Table 33 – Attributes of the ConnectedAP element

Attribute	Description
iedName	a name identifying the IED
apName	a name identifying this access point within the IED
desc	some descriptive text for this access point at this subnetwork

Each connected access point optionally has one server related address, and additional address information for real time communication related control blocks such as GSE control and SMV control. If all three are missing, it describes only the Subnetwork connection topology, for example for communication performance studies. For a complete SCD file, either the server address or at least one control block address shall be specified.

Further exists the optional element *PhysConn* describing one or more physical connections to this access point.

9.4.3 Address definition

The *Address* element contains the address parameters of this access point at this bus, at least one parameter. The different parameters are defined within the contained P elements. The *type* attribute of P identifies the meaning of the value. The meaning of the P parameters depend on the subnetwork protocol type and therefore has to be specified in the appropriate SCSM. Those used for IEC 61850-8-1 and IEC 61850-9-x are contained in the *type* enumeration type tPTypeEnum. For an explanation, see the appropriate standard parts.

```
<xs:complexType name="tAddress">
    <xs:sequence>
        <xs:element name="P" type="tP" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

The access point address shall be filled with a unique value at least for server type access points to get a complete SCD description.

```
<xs:complexType name="tP">
    <xs:simpleContent>
        <xs:extension base="tPAddr">
            <xs:attribute name="type" type="tPTypeEnum" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

tPAddr is a (non-empty) string containing no special characters such as LF, CR, or Tab. The pre-defined values for *tPTypeEnum* are as defined in IEC 61850-8-1. Custom-defined address types are also allowed (see below).

In order to be able to provide better validation of the address content by an XML parser, *tP* has been restricted (in the XML Schema sense) for each of these pre-defined address types. These type restrictions are named "*tP_*" followed by the address type as in *tPTypeEnum*. To use these restrictions, the *xsi:type* attribute must be given in the *P* element. Thus, there are two ways to provide such an address. For instance, for an IP address, both of the following formulations are equivalent from a syntactical and semantical point of view:

```
<P type="IP">10.0.0.11</P>
<P type="IP" xsi:type="tP_IP">10.0.0.11</P>
```

The advantage of the second, which uses the restriction type of *tP*, is that the address value (here "10.0.0.11") can also be validated by an XML parser. Using the first formulation, an address value of "abc" would be considered as perfectly valid, while the second formulation expects a value of the form "ddd.ddd.ddd.ddd", where each d corresponds to a digit.

Even if the restricted type is used, the (correct) address type must be specified.

Restrictions

- Extensions of the P *type* enumeration type *tPTypeEnum* shall start with a capital letter, and contain only alphanumeric characters and dashes(-),

9.4.4 GSE address definition

All control block address information is based on the abstract tControlBlock type. It provides the Address element for stating the control block related address parameters, and the reference to the control block within the IED by means of the *ldInst* and *cbName* attributes. Since GSE as well as SMV control blocks shall be located within LLN0, this is sufficient.

```
<xs:complexType name="tControlBlock" abstract="true">
    <xs:annotation>
        <xs:documentation xml:lang="en">A control block within a Logical Device (in LLN0).</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Address" type="tAddress" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="ldInst" type="tName" use="required"/>
            <xs:attribute name="cbName" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The GSE element defines the address for a GSE control block in this IED.

```
<xs:complexType name="tGSE">
    <xs:complexContent>
        <xs:extension base="tControlBlock">
            <xs:sequence>
                <xs:element name="MinTime" type="tDurationInMilliSec" minOccurs="0"/>
                <xs:element name="MaxTime" type="tDurationInMilliSec" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes have the following meaning as shown in Table 34.

Table 34 – Attributes of the GSE element

Attribute	Description
desc	Textual description
ldInst	The instance identification of the LD within this IED, on which the control block is located. An LN is not necessary, as these control blocks are only in LLN0.
cbName	The name of the control block within the LLN0 of the LD ldInst.

The Address element contains the GSE address parameters in the same syntax as the server address. The appropriate P *type* values are defined in the appropriate SCSMs.

The Mintime and Maxtime elements specify the following times:

Mintime — the maximal allowed sending delay on a data change in ms.

Maxtime — the source supervision time in ms (supervision heartbeat cycle time). Within this time, a failure of the source shall be detected by the client.

Mintime and Maxtime may influence SCSM parameters. Which parameters and how they are influenced is defined in the appropriate SCSM.

9.4.5 SMV address definition

The SMV element defines the address for a sampled value control block, like the GSE element does for the GSE control blocks. It is also based on the tControlBlock schema type, and therefore has the same attributes as the GSE control block.

```
<xs:complexType name="tSMV">
    <xs:complexContent>
        <xs:extension base="tControlBlock"/>
    </xs:complexContent>
</xs:complexType>
```

The attributes have the following meanings as shown in table 35.

Table 35 – Attributes of the SMV element

Attribute	Description
desc	Textual description.
ldInst	The instance identification of the LD within this IED, on which the control block is located. An LN is not necessary, as these control blocks are only in LLN0.
cbName	The name of the control block within the LLN0 of the LD ldInst.

The Address element contains the SMV address parameters in the same syntax as the server address. The appropriate P *type* values are defined in the appropriate SCSMs.

9.4.6 Physical connection parameters

The element PhysConn defines the type(s) of physical connection for this access point. The parameter values depend on the type of the physical connection, and their types (meaning) have to be defined in the stack mapping. Additional types may be introduced for documentation purpose.

```
<xs:complexType name="tPhysConn">
    <xs:sequence>
        <xs:element name="P" type="tP" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:normalizedString" use="required"/>
</xs:complexType>
```

The type attribute specifies the type of physical connection of this access point to the bus, while the value then specifies the instance of this type (for example type=”Plug”, value is “ST”). Allowed types and values shall be defined in the stack mapping. The P element can be repeated if one value is not sufficient. For the physical connections defined in IEC 61850-8-1, the types and corresponding values as shown in table 36 shall be used.

Table 36 – PhysConn P-Type definitions

PhysConn *type*	P *type*	Recommended values (IEC 61850-8-1 related)
Connection	Type	10BaseT for electrical connection FOC for optical connection Radio for radio connection, for example WLAN
	Plug	RJ45 for electrical plug ST for bajonet plug (optical glass)

Restrictions

- The PhysConn *type* values shall start with a capital letter, and contain only alphanumeric characters.

9.4.7 Communication section example

The following SCL part shows a communication section with one subnetwork XW1, to which two IEDs are connected with their access points S1. The protocol type 8-MMS specifies a protocol as defined in IEC 61850-8-1 and IEC 61850-9-2. The PhysConn and address types are just examples. One IED also contains a GSE control block with an address, however without the MaxTime and MinTime elements, which are optional. Another contains a sampled value control block.

```
<Communication>
    <SubNetwork name="W01" type="8-MMS">
        <Text>Station bus</Text>
        <BitRate unit="b/s">10</BitRate>
        <ConnectedAP iedName="D1Q1SB4" apName="S1">
            <Address>
                <P type="IP">10.0.0.11</P>
                <P type="IP-SUBNET">255.255.255.0</P>
                <P type="IP-GATEWAY">10.0.0.101</P>
                <P type="OSI-TSEL">00000001</P>
                <P type="OSI-PSEL">01</P>
                <P type="OSI-SSEL">01</P>
            </Address>
            <PhysConn type="Plug">
                <P type="type">FOC</P>
                <P type="Plug">ST</P>
            </PhysConn>
            <SMV ldInst="C1" cbName="Volt">
                <Address>
                    <P type="MAC-Address">01-0C-CD-04-00-01</P>
                    <P type="APPID">4000</P>
                    <P type="VLAN-ID">123</P>
                    <P type="VLAN-PRIORITY">4</P>
                </Address>
            </SMV>
        </ConnectedAP>
        <ConnectedAP iedName="E1Q1SB1" apName="S1">
            <Address>
                <P type="IP">10.0.0.1</P>
                <P type="IP-SUBNET">255.255.255.0</P>
                <P type="IP-GATEWAY">10.0.0.101</P>
                <P type="OSI-TSEL">00000001</P>
                <P type="OSI-PSEL">01</P>
                <P type="OSI-SSEL">01</P>
            </Address>
            <GSE ldInst="C1" cbName="Goose1">
                <Address>
                    <P type="MAC-Address">01-0C-CD-01-00-01</P>
                    <P type="APPID">3000</P>
                    <P type="VLAN-PRIORITY">4</P>
                </Address>
            </GSE>
        </ConnectedAP>
    </SubNetwork>
</Communication>
```

9.5 Data type templates

9.5.1 General

This Clause defines instantiable logical node types. A logical node type is an instantiable template of the data of a logical node. A LNodeType (elsewhere also called LN type) is referenced each time that this instantiable type is needed within an IED. A logical node type template is built from DATA (DO) elements, which again have a DO type, which is derived from

the DATA classes (CDC) defined in IEC 61850-7-3. DOs consist of attributes (DA) or of elements of already defined DO types (SDO). The attribute (DA) has a functional constraint, and can either have a basic type, be an enumeration, or a structure of a DAType. The DAType is built from BDA elements, defining the structure elements, which again can be BDA elements or have a base type such as a DA.

All types are uniquely identified by their type *id*, and by an *iedType* attribute. On generation of the system SCD file from IED ICD files, the LN type identifications may have to change to keep uniqueness across all IED definitions. To keep possible semantic information of the type names, it is recommended to use the *iedType* attribute to define the relation of a specific LN type to an IED type. If this is not sufficient, a new LN type name can be generated by concatenating the IED name (which shall be unique within the file) with the old type name (which shall be unique at least per IED). If a LN type is generally valid for several IEDs, then the IEDType attribute shall be defined as empty string. This is especially necessary for type definitions which shall be used in an SSD file, where no IEDs and therefore no IED types exist.

The order of DO elements within a LNodeType definition, and of SDO/DA elements within a DOType definition shall also specify the order of data values within a message, if this is not specified elsewhere, for example by explicit FCDA definitions in a data set down to the attribute. The order in the LNodeType definition is the responsibility of the IED configurator tool, while the order in the data set is the responsibility of the system configurator tool.

The following UML figure gives an overview of the DataTypeTemplate section of the Schema.

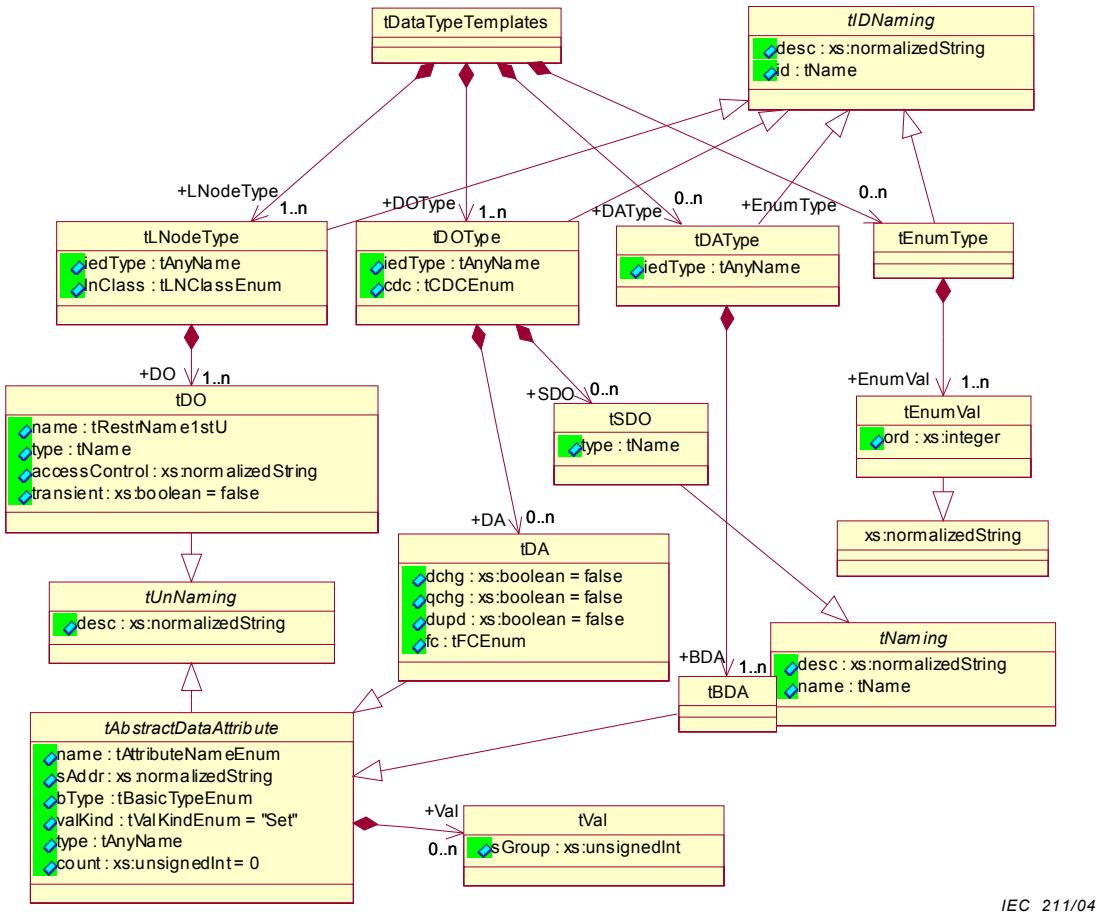


Figure 17 – UML overview of DataTypeTemplate section

Here follows the XML schema definition inclusive defined restrictions within *DataTypeTemplates*.

```
<xs:element name="DataTypeTemplates" type="tDataTypeTemplates">
    <xs:unique name="uniqueLNodeType">
        <xs:selector xpath="scl:LNodeType"/>
        <xs:field xpath="@id"/>
        <xs:field xpath="@iedType"/>
    </xs:unique>
    <xs:key name="DOTypeKey">
        <xs:selector xpath="scl:DOType"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:keyref name="ref2DOType" refer="DOTypeKey">
        <xs:selector xpath="scl:LNodeType/scl:DO"/>
        <xs:field xpath="@type"/>
    </xs:keyref>
    <xs:keyref name="ref2DOTypeForSDO" refer="DOTypeKey">
        <xs:selector xpath="scl:DOType/scl:SDO"/>
        <xs:field xpath="@type"/>
    </xs:keyref>
    <xs:key name="DATypeKey">
        <xs:selector xpath="scl:DAType"/>
        <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="EnumTypeKey">
        <xs:selector xpath="scl:EnumType"/>
        <xs:field xpath="@id"/>
    </xs:key>

<xs:complexType name="tDataTypeTemplates">
    <xs:sequence>
        <xs:element name="LNodeType" type="tLNodeType" maxOccurs="unbounded">
            <xs:unique name="uniqueDOInLNodeType">
                <xs:selector xpath="scl:DO"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="DOType" type="tDOType" maxOccurs="unbounded">
            <xs:unique name="uniqueDAorSDOInLDOType">
                <xs:selector xpath="./*"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="DAType" type="tDAType" minOccurs="0" maxOccurs="unbounded">
            <xs:unique name="uniqueBDAInLDAType">
                <xs:selector xpath="scl:BDA"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="EnumType" type="tEnumType" minOccurs="0" maxOccurs="unbounded">
            <xs:unique name="uniqueOrdInEnumType">
                <xs:selector xpath="scl:EnumVal"/>
                <xs:field xpath="@ord"/>
            </xs:unique>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

In SCL, all types are contained in the DataTypeTemplates section. As can be seen by the schema part above, the type definitions shown in Table 37 can appear there.

Table 37 – Template definition elements

Element name of Template part	Description
LNodeType	An instantiable logical node type, as referenced from IEDs and from the Substation section, and as defined in IEC 61850-7-4
DOType	An instantiable DATA type; referenced from LNodeType or from the SDO element of another DOType. Instantiable version based on the CDC definitions from IEC 61850-7-3.
DAType	An instantiable structured attribute type; referenced from within a DA element of a DOType, or from within another DAType for nested type definitions. Based on the attribute structure definitions of IEC 61850-7-3.
EnumType	An enumeration type; referenced from the DA element of a DOType or from a DAType, in case that the *bType* is Enum. The definitions shall follow enumeration definitions from IEC 61850-7-3 and IEC 61850-7-4.

9.5.2 LNodeType definitions

The LN type (LNodeType element) contains a list of DATA (Data objects, DO), its attributes, and possible default values for configuration parameters.

```
<xs:complexType name="tLNodeType">
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:sequence>
                <xs:element name="DO" type="tDO" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="iedType" type="tAnyName" use="optional"/>
            <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes have the following meaning as shown in Table 38.

Table 38 – Attributes of the LNodeType element

Attribute	Description
id	A reference identifying this LN type within this SCL section; used by the LN attribute LNType to reference this definition
desc	An additional text describing this LN type
iedType	The manufacturer IED type of the IED to which this LN type belongs
lnClass	The LN base class of this type as specified in IEC 61850-7-3; observe that here an enumeration exists, which allows extensions (names containing only capital letters)

The DO element references the instantiable data type of this DO.

```
<xs:complexType name="tDO">
    <xs:annotation>
        <xs:documentation xml:lang="en">See Section 9.5.1</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:attribute name="name" type="tRestrName1stU" use="required"/>
            <xs:attribute name="type" type="tName" use="required"/>
            <xs:attribute name="accessControl" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="transient" type="xs:boolean" use="optional" default="false"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The DO attributes are used as shown in Table 39.

Table 39 – Attributes of the DO element

Attribute	Description
name	The DATA name as specified for example in IEC 61850-7-4.
type	The *type* references the *id* of a DOType definition.
accessControl	Access control definition for this DO. If it is missing then any higher-level access control definition applies.
transient	If set to true, it indicates that the Transient definition from IEC 61850-7-4 applies.

9.5.3 DO type definition

The DOType element referenced by the *type* attribute of the LNodeType DO element has the following syntax:

```
<xs:complexType name="tDOType">
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:choice minOccurs="1" maxOccurs="unbounded">
                <xs:element name="SDO" type="tSDO"/>
                <xs:element name="DA" type="tDA"/>
            </xs:choice>
            <xs:attribute name="iedType" type="tAnyName" use="optional"/>
            <xs:attribute name="cdc" type="tCDCEnum" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The DOType identifies the contents of the DO. This can be either attributes (DA elements), or the reference to another DOType (SDO element). The attributes have the following meaning as shown in Table 40.

Table 40 – Attributes of the DOType element

Attribute	Description
id	The (global) identification of this DOType within an *iedType*. Used to reference this type.
iedType	The type of the IED to which this DOType belongs. The empty string allows references for all IED types, or from the Substation section.
cdc	The basic CDC (Common Data Class) as defined in IEC 61850-7-3.

The SDO element then references another DOType definition. Warning: recursive references are not allowed, can however not be checked at syntax level!

```
<xs:complexType name="tSDO">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:attribute name="type" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tDA">
```

The attributes of the SDO element are defined in Table 41.

Table 41 – Attributes of the SDO element

Attribute	Description
name	The SDO name
desc	Descriptive text for the SDO
type	References the DOType defining the contents of the SDO

The attribute (DA) definition carries the handling attributes according to IEC 61850-7-3 as defined in the appropriate tables. Each instantiable atribute shall be defined in the DO type definition. Observe that a certain SCSM (for example IEC 61850-8-1) might define additional mandatory attributes or SDOs. The DA syntax is described in the next Subclause.

9.5.4 Data attribute (DA) definition

9.5.4.1 General

The DA element defines the attributes, their stack related handling, and describes their (default) values, if there are any known.

The DA element has either a basic type, or again a reference to a structured attribute type definition for example in the case of an attribute with a structure such as *ScaledValueConfig*. If the DA is an array, then its count attribute gives the number of array elements. IEC 61850-7-3 and for some enumerations IEC 61850-7-4 define the type of a certain attribute based on the CDC of the DO.

The value coding syntax in the Val element of the DA element then has to follow the XML schema data type coding definitions for the IEC 61850-7-x basic data types. The type mapping is as shown in Table 42.

Table 42 – Data type mapping

IEC 61850-7-x basic type	XML Schema (xs) data type	Value representation
INT8, INT16, INT24, INT32, INT8U, INT16U, INT24U, INT32U	integer	An integer number, no decimal fraction (99999)
FLOAT32, FLOAT64	double	A number with or without a decimal fraction (999,99999).
BOOLEAN	boolean	*false*, *true* or 0, 1
ENUMERATED, CODED ENUM	normalizedString	The enumeration element names as defined in IEC 61850-7-x as string values
Octet string	base64Binary	Coding according to 6.8 of RFC 2045
VisibleString	normalizedString	A character string without tabs, linefeeds and carriage return, restricted to 8-bit characters (UTF-8 single byte coding, ISO/IEC 8859-1)
UnicodeString	normalizedString	A character string without tabs, linefeeds and carriage return. All characters in an XML file are principally Unicode, for example in UTF-8 coding

NOTE It is not intended to specify values of types Timestamp, EntryTime, INT128 and Quality in an SCL file, as these only belong to live process data.

The meaning of the value for an IED configuration tool can be different depending on the device capabilities, the functional characteristic of the attribute, and the stage of the engineering process. The DA attribute *valKind* allows the specification of this meaning. It is ignored if no value is given, and for all cases not specified in Table 43 (for example for the *q* and *t* attributes).

```
<xs:simpleType name="tValKindEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="Spec"/>
        <xs:enumeration value="Conf"/>
        <xs:enumeration value="RO"/>
        <xs:enumeration value="Set"/>
    </xs:restriction>
</xs:simpleType>
```

Table 43 – Attribute value kind (Valkind) meaning

Valkind value	Functional constraints	Engineering process stage	Meaning
Spec	Non operational (CF, DC)	Specification phase	The wanted value determined at specification phase typically in an SCD file
Conf	CF, DC, operational attribute of a CDC used for settings	IED template, after IED engineering	This value is not available online at the IED. The IED is engineered such that this value is used
RO	Operational process state attribute	IED template	The default value for the attribute to be used if *q.source* is set to *defaulted* or if the value is fix on the IED
RO	CF, DC, operational attribute of data used for settings	IED template, after IED configuration	Read only value at an IED – can only be set at configuration time
Set	CF, DC	At/after IED configuration	A determined setting value. The value is/shall be set within the IED
Set	Operational process values (except time and quality)	At/after IED configuration (possibly RO changed to Set)	The default value for the operational attribute, to be used if *q.source* is set to *defaulted*
Set	Operational setting value (SP, SG for all data used as setting)	At/after IED configuration	The setting value for the set point respectively parameter

This allows, for example, the definition of IED capabilities (which attributes are available, which are read only), the default values an IED is delivered with (readable, changeable, or not visible at all), or the setting values for operative (for example protection) parameters.

Here follows the syntax definition. It is based on an abstract type tAbstractDatAttribute which is reused later in attribute structure definitions.

```
<xs:complexType name="tDA">
    <xs:complexContent>
        <xs:extension base="tAbstractDataAttribute">
            <xs:attributeGroup ref="agDATrgOp"/>
            <xs:attribute name="fc" type="tFCEnum" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:attributeGroup name="agDATrgOp">
    <xs:attribute name="dchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="qchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dupd" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>

<xs:complexType name="tAbstractDataAttribute" abstract="true">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Val" type="tVal" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" type="tAttributeNameEnum" use="required"/>
            <xs:attribute name="sAddr" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="bType" type="tBasicTypeEnum" use="required"/>
            <xs:attribute name="valKind" type="tValKindEnum" use="optional" default="Set"/>
            <xs:attribute name="type" type="tAnyName" use="optional"/>
            <xs:attribute name="count" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The attributes of the DA element are defined in Table 44.

Table 44 – Attributes of the DA element

Attribute	Description
desc	Some descriptive text for the attribute.
name	The attribute name; the type tAttributeEnum restricts to the attribute names from IEC 61850-7-3, plus new ones starting with lower case letters.
fc	The functional constraint for this attribute; fc=SG always also implies fc=SE; if an attribute has ST and CO respectively MX and SP, then always the fc value of the status has to be taken. The second fc is either defined by SCSM related attributes (for example in IEC 61850-8-1), or implied by the ctlModel values.
dchg, qchg, dupd	Defines which trigger options are supported by the attribute (value true means supported).
sAddr	an optional short address of this DO attribute (see 9.5.4.3).
bType	The basic type of the attribute, taken from tBasicTypeEnum (see 9.5.4.2).
type	Only used if *bType*= Enum or *bType* = Struct to refer to the appropriate enumeration type or DAType (attribute structure) definition.
count	Optional. Shall state the number of array elements in case that the attribute is an array.
valKind	Determines how the value shall be interpreted if any is given – see Table 43.

The attributes *name, fc*, and *bType* shall always be defined. All instantiable attributes contained within a DO shall be defined.

9.5.4.2 Attribute basic types

The basic types allowed are as follows:

```
<xs:simpleType name="tPredefinedBasicTypeEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="BOOLEAN"/>
        <xs:enumeration value="INT8"/>
        <xs:enumeration value="INT16"/>
        <xs:enumeration value="INT24"/>
        <xs:enumeration value="INT32"/>
        <xs:enumeration value="INT128"/>
        <xs:enumeration value="INT8U"/>
        <xs:enumeration value="INT16U"/>
        <xs:enumeration value="INT24U"/>
        <xs:enumeration value="INT32U"/>
        <xs:enumeration value="FLOAT32"/>
        <xs:enumeration value="FLOAT64"/>
        <xs:enumeration value="Enum"/>
        <xs:enumeration value="Dbpos"/>
        <xs:enumeration value="Tcmd"/>
        <xs:enumeration value="Quality"/>
        <xs:enumeration value="Timestamp"/>
        <xs:enumeration value="VisString32"/>
        <xs:enumeration value="VisString64"/>
        <xs:enumeration value="VisString255"/>
        <xs:enumeration value="Octet64"/>
        <xs:enumeration value="Struct"/>
        <xs:enumeration value="EntryTime"/>
        <xs:enumeration value="Unicode255"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tExtensionBasicTypeEnum">
    <xs:annotation>
        <xs:documentation xml:lang="en">User extensible basic types.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:Name">
        <xs:pattern value="[\p{L},\d]+"/>
    </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="tBasicTypeEnum">
    <xs:annotation>
        <xs:documentation xml:lang="en">All possible basic types.</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="tPredefinedBasicTypeEnum tExtensionBasicTypeEnum"/>
</xs:simpleType>
```

tPredefinedBasicTypeEnum contains the definitions as defined in IEC 61850-7-x. CODED ENUMs are replace by concrete basic types Quality, Dbpos for double bit positions as used in DPC and DPS, and Tcmd for tap changer commands used in BSC. While Quality remains opaque (no values required in SCL), Dbpos and Tcmd are handled like Enums for the value coding. For VisibleString, UnicodeString and OctetString length dependent (sub-)types are introduced. VisString32 is for example a VisibleString of maximum length of 32 characters.

tBasicTypeEnum allows the extension of the basic types according to the stated rule, i.e. the first character shall be a capital letter, the rest may be arbitrary alphanumeric characters, at least one. This extension facility is foreseen for other application area standards and SCSMs, not for private use.

The following example defines the stVal attribute of a DPC CDC without value, according to IEC 61850-7-3:

```
<DA name="stVal" fc="ST" dchg="true" bType="Dbpos"/>
```

9.5.4.3 Short addresses

The *sAddr* attribute allows the allocation of a short address to DO attributes. Short addresses can be used within the communication to be more efficient either in the communication, or in the handling of messages at client or server. Furthermore, they can be used as IED internal identification for the attribute. To be able to use short addresses in the communication,

- the stack mapping must allow them and define their meaning, and
- the IED must allow them.

The detailed syntax of a short address value depends on the stack if the stack (SCSM) defines their usage, or else on the IED tool. SCL foresees a two level hierarchy for short addresses used in communication:

1) The communication address of the IED/server/access point.
2) The short address of a data item at attribute level.

It is possible to use the short address instead of the (symbolic) IED communication address if the short address is unique system-wide, and the stack (SCSM) allows this. Otherwise, the short address value scope and syntax is private to the IED.

Tools which do not handle short addresses shall also preserve imported contents in exported SCL files.

9.5.4.4 Values

The optional value definition contains one value. For attributes with *fc* = **SG**, the *sGroup* attribute specifies to which setting group this value belongs. There may be a value for each defined setting group. The meaning of the value in the engineering process is defined at the DA/DAI level by means of the *valKind* attribute.

```
<xs:complexType name="tVal">
    <xs:simpleContent>
        <xs:extension base="xs:normalizedString">
            <xs:attribute name="sGroup" type="xs:unsignedInt" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Attribute description

sGroup the number of the setting group (if *fc* = "**SG**") to which this value belongs.

The *sGroup* value used within an IED should be checked against an existing setting group definition on this IED, where the maximum allowed number is specified (SettingControl.numOfSGs). If the optional *sGroup* attribute is missing completely, then either the concerned DATA attribute is in no setting group (*fc* # **SG**), or the data value applies to all setting groups.

9.5.5 Data attribute structure type

In case the DA.*bType* value is **Struct**, the DA.*type* attribute references an attribute structure. These structures are defined with DAType elements.

```
<xs:complexType name="tDAType">
    <xs:annotation>
        <xs:documentation xml:lang="en">See Section 9.5.2</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:sequence>
                <xs:element name="BDA" type="tBDA" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="iedType" type="tAnyName" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The DAType element contains a list of attributes with the BDA element. These attributes can either have a basic type, or refer to another attribute structure. The definitions have to follow IEC 61850-7-3 in structure, type and naming.

```
<xs:complexType name="tBDA">
    <xs:annotation>
        <xs:documentation xml:lang="en">Basic Data Attribute?</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tAbstractDataAttribute"/>
    </xs:complexContent>
</xs:complexType>
```

The BDA element instantiates the tAbstractDataAttribute and has therefore the same attributes.

The attributes of the BDA element are defined in Table 45.

Table 45 – Attributes of the BDA element

Attribute	Description
desc	Some descriptive text for the attribute.
name	The attribute name; the type tAttributeEnum restricts to the attribute names from IEC 61850-7-3, plus new ones starting with lower case letters.
sAddr	an optional short address of this BDA attribute.
bType	The basic type of the attribute, taken from tBasicTypeEnum.
type	Only used if *bType*= Enum or *bType* = Struct to refer to the appropriate enumeration type or DAType definition.
count	Optional. Shall state the number of array elements in the case where the attribute is an array.
valKind	Determines how the value shall be interpreted if any is given – see Table 43.

Note that the sAddr attribute might appear on several levels, starting with the DA element. There are in principle two methods to handle this:

1) Use only the lowest level value.
2) Use values on all levels as a kind of hierarchical short address.

It is up to the SCSM respectively the IED tool to decide which method is used (see also 9.5.4.3).

For *valKind* only the lowest level value shall be used.

9.5.6 Enumeration types

Enumerations are in general used in more than one LNodeType. Therefore, an enumeration type definition is made for them.

```
<xs:complexType name="tEnumType">
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:sequence>
                <xs:element name="EnumVal" type="tEnumVal" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Enumeration definitions are valid for all IEDs; they are not IED type dependent. Therefore the allowed names are standardized as follows:

- For enumerations from IEC 61850-7-3, the name of the attribute shall be taken. In the case where for different CDCs the same attribute name is used for different enumerations, the CDC name shall be used additionally in front of the attribute name.
- Enumerations from IEC 61850-7-4 are defined on top of INC or INS common data classes. Therefore both the status value and (for INC) the control value shall have Enum type instead of INT32. Also on stack level the mappings for Enum data types shall apply. For these enumerations the name of the DATA shall be taken. In case that for different LN classes the same DATA name is taken for different enumerations, then the following cases apply:
  - — one enumeration is a subset of the other: in this case the superset shall be used as enumeration,
  - — the enumerations are different: then the LN class name shall be used additionally in front of the DATA name.

The resulting normative enumeration definitions from IEC 61850-7-3 and IEC 61850-7-4 are listed in Annex B. They also serve as examples for enumeration definitions.

If the semantics of the same LN class code and same DATA name code for an enumeration in another IEC name space is redefined, then the enumeration type and its values shall also be kept unchanged (possibly with redefined semantics or with value extensions).

The meaning of the attributes of the EnumType element is as shown in Table 46.

Table 46 – Attributes of the EnumType element

Attribute	Description
id	A reference identifying this enumeration type; used by the *type* attribute of DA and BDA elements to reference this definition in the case where the *bType* is Enum.
desc	An additional text describing this LN type.

The values of the enumeration are defined as follows:

```
<xs:complexType name="tEnumVal">
    <xs:simpleContent>
        <xs:extension base="xs:normalizedString">
            <xs:attribute name="ord" type="xs:integer" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The *ord* attribute contains the order of the values, starting from 0. The value of type normalizedString is the character string as defined in IEC 61850-7-3 or IEC 61850-7-4.

9.5.7 Data type template examples

Examples can be found in the *DataTypeTemplate* section of Clause D.2.

Annex A
(normative)

SCL syntax: XML schema definition

A.1 Base types

File **SCL_BaseSimpleTypes.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="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"
version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <xs:simpleType name="tRef">
        <xs:restriction base="xs:normalizedString">
            <xs:pattern value=".+/.+/.+/.+"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tAnyName">
        <xs:restriction base="xs:normalizedString"/>
    </xs:simpleType>
    <xs:simpleType name="tName">
        <xs:restriction base="tAnyName">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tRestrName">
        <xs:restriction base="xs:Name">
            <xs:pattern value="[\d,\p{L}]+"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tRestrName1stU">
        <xs:restriction base="xs:Name">
            <xs:pattern value="\p{Lu}[\d,\p{L}]*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tRestrName1stL">
        <xs:restriction base="xs:Name">
            <xs:pattern value="\p{Ll}[\d,\p{L}]*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tPAddr">
        <xs:restriction base="xs:normalizedString">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

File **SCL_Enums.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="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"
version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_BaseSimpleTypes.xsd"/>
    <xs:simpleType name="tPredefinedPTypeEnum">
        <xs:restriction base="xs:Name">
```

```
            <xs:enumeration value="IP"/>
            <xs:enumeration value="IP-SUBNET"/>
            <xs:enumeration value="IP-GATEWAY"/>
            <xs:enumeration value="OSI-NSAP"/>
            <xs:enumeration value="OSI-TSEL"/>
            <xs:enumeration value="OSI-SSEL"/>
            <xs:enumeration value="OSI-PSEL"/>
            <xs:enumeration value="OSI-AP-Title"/>
            <xs:enumeration value="OSI-AP-Invoke"/>
            <xs:enumeration value="OSI-AE-Qualifier"/>
            <xs:enumeration value="OSI-AE-Invoke"/>
            <xs:enumeration value="MAC-Address"/>
            <xs:enumeration value="APPID"/>
            <xs:enumeration value="VLAN-PRIORITY"/>
            <xs:enumeration value="VLAN-ID"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionPTypeEnum">
        <xs:restriction base="xs:normalizedString">
            <xs:pattern value="\p{Lu}[\d,\p{L},\-]*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tPTypeEnum">
        <xs:union memberTypes="tPredefinedPTypeEnum tExtensionPTypeEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedAttributeNameEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="T"/>
            <xs:enumeration value="Test"/>
            <xs:enumeration value="Check"/>
            <xs:enumeration value="SIUnit"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionAttributeNameEnum">
        <xs:restriction base="tRestrName1stL"/>
    </xs:simpleType>
    <xs:simpleType name="tAttributeNameEnum">
        <xs:union memberTypes="tPredefinedAttributeNameEnum tExtensionAttributeNameEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedCommonConductingEquipmentEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="CBR"/>
            <xs:enumeration value="DIS"/>
            <xs:enumeration value="VTR"/>
            <xs:enumeration value="CTR"/>
            <xs:enumeration value="GEN"/>
            <xs:enumeration value="CAP"/>
            <xs:enumeration value="REA"/>
            <xs:enumeration value="CON"/>
            <xs:enumeration value="MOT"/>
            <xs:enumeration value="EFN"/>
            <xs:enumeration value="PSH"/>
            <xs:enumeration value="BAT"/>
            <xs:enumeration value="BSH"/>
            <xs:enumeration value="CAB"/>
            <xs:enumeration value="GIL"/>
            <xs:enumeration value="LIN"/>
            <xs:enumeration value="RRC"/>
            <xs:enumeration value="SAR"/>
            <xs:enumeration value="TCF"/>
            <xs:enumeration value="TCR"/>
            <xs:enumeration value="IFL"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionEquipmentEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="E\p{Lu}*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tCommonConductingEquipmentEnum">
        <xs:union memberTypes="tPredefinedCommonConductingEquipmentEnum tExtensionEquipmentEnum"/>
    </xs:simpleType>
```

```
    <xs:simpleType name="tPowerTransformerEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="PTR"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tTransformerWindingEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="PTW"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedEquipmentEnum">
        <xs:union memberTypes="tCommonConductingEquipmentEnum tPowerTransformerEnum
tTransformerWindingEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tEquipmentEnum">
        <xs:union memberTypes="tPredefinedEquipmentEnum tExtensionEquipmentEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedGeneralEquipmentEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="AXN"/>
            <xs:enumeration value="BAT"/>
            <xs:enumeration value="MOT"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionGeneralEquipmentEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="E\p{Lu}*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tGeneralEquipmentEnum">
        <xs:union memberTypes="tPredefinedGeneralEquipmentEnum tExtensionGeneralEquipmentEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tServiceSettingsEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="Dyn"/>
            <xs:enumeration value="Conf"/>
            <xs:enumeration value="Fix"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tPhaseEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="A"/>
            <xs:enumeration value="B"/>
            <xs:enumeration value="C"/>
            <xs:enumeration value="N"/>
            <xs:enumeration value="all"/>
            <xs:enumeration value="none"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tAuthenticationEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="none"/>
            <xs:enumeration value="password"/>
            <xs:enumeration value="week"/>
            <xs:enumeration value="strong"/>
            <xs:enumeration value="certificate"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tAssociationKindEnum">
        <xs:restriction base="xs:token">
            <xs:enumeration value="pre-established"/>
            <xs:enumeration value="predefined"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tLPHDEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="LPHD"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tLLN0Enum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="LLN0"/>
```

```
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupAEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="A[A-Z]*"/>
        <xs:enumeration value="ANCR"/>
        <xs:enumeration value="ARCO"/>
        <xs:enumeration value="ATCC"/>
        <xs:enumeration value="AVCO"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupCEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="C[A-Z]*"/>
        <xs:enumeration value="CILO"/>
        <xs:enumeration value="CSWI"/>
        <xs:enumeration value="CALH"/>
        <xs:enumeration value="CCGR"/>
        <xs:enumeration value="CPOW"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupGEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="G[A-Z]*"/>
        <xs:enumeration value="GAPC"/>
        <xs:enumeration value="GGIO"/>
        <xs:enumeration value="GSAL"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupIEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="I[A-Z]*"/>
        <xs:enumeration value="IHMI"/>
        <xs:enumeration value="IARC"/>
        <xs:enumeration value="ITCI"/>
        <xs:enumeration value="ITMI"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupMEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="M[A-Z]*"/>
        <xs:enumeration value="MMXU"/>
        <xs:enumeration value="MDIF"/>
        <xs:enumeration value="MHAI"/>
        <xs:enumeration value="MHAN"/>
        <xs:enumeration value="MMTR"/>
        <xs:enumeration value="MMXN"/>
        <xs:enumeration value="MSQI"/>
        <xs:enumeration value="MSTA"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDomainLNGroupPEnum">
    <xs:restriction base="xs:Name">
        <xs:pattern value="P[A-Z]*"/>
        <xs:enumeration value="PDIF"/>
        <xs:enumeration value="PDIS"/>
        <xs:enumeration value="PDIR"/>
        <xs:enumeration value="PDOP"/>
        <xs:enumeration value="PDUP"/>
        <xs:enumeration value="PFRC"/>
        <xs:enumeration value="PHAR"/>
        <xs:enumeration value="PHIZ"/>
        <xs:enumeration value="PIOC"/>
        <xs:enumeration value="PMRI"/>
        <xs:enumeration value="PMSS"/>
        <xs:enumeration value="POPF"/>
        <xs:enumeration value="PPAM"/>
        <xs:enumeration value="PSCH"/>
        <xs:enumeration value="PSDE"/>
        <xs:enumeration value="PTEF"/>
        <xs:enumeration value="PTOC"/>
        <xs:enumeration value="PTOF"/>
```

```
            <xs:enumeration value="PTOV"/>
            <xs:enumeration value="PTRC"/>
            <xs:enumeration value="PTTR"/>
            <xs:enumeration value="PTUC"/>
            <xs:enumeration value="PTUV"/>
            <xs:enumeration value="PUPF"/>
            <xs:enumeration value="PTUF"/>
            <xs:enumeration value="PVOC"/>
            <xs:enumeration value="PVPH"/>
            <xs:enumeration value="PZSU"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupREnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="R[A-Z]*"/>
            <xs:enumeration value="RSYN"/>
            <xs:enumeration value="RDRE"/>
            <xs:enumeration value="RADR"/>
            <xs:enumeration value="RBDR"/>
            <xs:enumeration value="RDRS"/>
            <xs:enumeration value="RBRF"/>
            <xs:enumeration value="RDIR"/>
            <xs:enumeration value="RFLO"/>
            <xs:enumeration value="RPSB"/>
            <xs:enumeration value="RREC"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupSEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="S[A-Z]*"/>
            <xs:enumeration value="SARC"/>
            <xs:enumeration value="SIMG"/>
            <xs:enumeration value="SIML"/>
            <xs:enumeration value="SPDC"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupTEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="T[A-Z]*"/>
            <xs:enumeration value="TCTR"/>
            <xs:enumeration value="TVTR"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupXEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="X[A-Z]*"/>
            <xs:enumeration value="XCBR"/>
            <xs:enumeration value="XSWI"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupYEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="Y[A-Z]*"/>
            <xs:enumeration value="YPTR"/>
            <xs:enumeration value="YEFN"/>
            <xs:enumeration value="YLTC"/>
            <xs:enumeration value="YPSH"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNGroupZEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="Z[A-Z]*"/>
            <xs:enumeration value="ZAXN"/>
            <xs:enumeration value="ZBAT"/>
            <xs:enumeration value="ZBSH"/>
            <xs:enumeration value="ZCAB"/>
            <xs:enumeration value="ZCAP"/>
            <xs:enumeration value="ZCON"/>
            <xs:enumeration value="ZGEN"/>
            <xs:enumeration value="ZGIL"/>
            <xs:enumeration value="ZLIN"/>
            <xs:enumeration value="ZMOT"/>
```

```
            <xs:enumeration value="ZREA"/>
            <xs:enumeration value="ZRRC"/>
            <xs:enumeration value="ZSAR"/>
            <xs:enumeration value="ZTCF"/>
            <xs:enumeration value="ZTCR"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tDomainLNEnum">
        <xs:union memberTypes="tDomainLNGroupAEnum tDomainLNGroupCEnum tDomainLNGroupGEnum
tDomainLNGroupIEnum tDomainLNGroupMEnum tDomainLNGroupPEnum tDomainLNGroupREnum
tDomainLNGroupSEnum tDomainLNGroupTEnum tDomainLNGroupXEnum tDomainLNGroupYEnum
tDomainLNGroupZEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedLNClassEnum">
        <xs:union memberTypes="tLPHDEnum tLLN0Enum tDomainLNEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tExtensionLNClassEnum">
        <xs:restriction base="xs:Name">
            <xs:minLength value="1"/>
            <xs:pattern value="\p{Lu}+"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tLNClassEnum">
        <xs:union memberTypes="tPredefinedLNClassEnum tExtensionLNClassEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedCDCEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="SPS"/>
            <xs:enumeration value="DPS"/>
            <xs:enumeration value="INS"/>
            <xs:enumeration value="ACT"/>
            <xs:enumeration value="ACD"/>
            <xs:enumeration value="SEC"/>
            <xs:enumeration value="BCR"/>
            <xs:enumeration value="MV"/>
            <xs:enumeration value="CMV"/>
            <xs:enumeration value="SAV"/>
            <xs:enumeration value="WYE"/>
            <xs:enumeration value="DEL"/>
            <xs:enumeration value="SEQ"/>
            <xs:enumeration value="HMV"/>
            <xs:enumeration value="HWYE"/>
            <xs:enumeration value="HDEL"/>
            <xs:enumeration value="SPC"/>
            <xs:enumeration value="DPC"/>
            <xs:enumeration value="INC"/>
            <xs:enumeration value="BSC"/>
            <xs:enumeration value="ISC"/>
            <xs:enumeration value="APC"/>
            <xs:enumeration value="SPG"/>
            <xs:enumeration value="ING"/>
            <xs:enumeration value="ASG"/>
            <xs:enumeration value="CURVE"/>
            <xs:enumeration value="DPL"/>
            <xs:enumeration value="LPL"/>
            <xs:enumeration value="CSD"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionCDCEnum">
        <xs:restriction base="xs:Name">
            <xs:minLength value="1"/>
            <xs:pattern value="\p{Lu}+"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tCDCEnum">
        <xs:union memberTypes="tPredefinedCDCEnum tExtensionCDCEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tTrgOptEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="dchg"/>
            <xs:enumeration value="qchg"/>
            <xs:enumeration value="dupd"/>
```

```
            <xs:enumeration value="none"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tTrgOptControlEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="dchg"/>
            <xs:enumeration value="qchg"/>
            <xs:enumeration value="dupd"/>
            <xs:enumeration value="period"/>
            <xs:enumeration value="none"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tFCEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="ST"/>
            <xs:enumeration value="MX"/>
            <xs:enumeration value="CO"/>
            <xs:enumeration value="SP"/>
            <xs:enumeration value="SG"/>
            <xs:enumeration value="SE"/>
            <xs:enumeration value="SV"/>
            <xs:enumeration value="CF"/>
            <xs:enumeration value="DC"/>
            <xs:enumeration value="EX"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tPredefinedBasicTypeEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="BOOLEAN"/>
            <xs:enumeration value="INT8"/>
            <xs:enumeration value="INT16"/>
            <xs:enumeration value="INT24"/>
            <xs:enumeration value="INT32"/>
            <xs:enumeration value="INT128"/>
            <xs:enumeration value="INT8U"/>
            <xs:enumeration value="INT16U"/>
            <xs:enumeration value="INT24U"/>
            <xs:enumeration value="INT32U"/>
            <xs:enumeration value="FLOAT32"/>
            <xs:enumeration value="FLOAT64"/>
            <xs:enumeration value="Enum"/>
            <xs:enumeration value="Dbpos"/>
            <xs:enumeration value="Tcmd"/>
            <xs:enumeration value="Quality"/>
            <xs:enumeration value="Timestamp"/>
            <xs:enumeration value="VisString32"/>
            <xs:enumeration value="VisString64"/>
            <xs:enumeration value="VisString255"/>
            <xs:enumeration value="Octet64"/>
            <xs:enumeration value="Struct"/>
            <xs:enumeration value="EntryTime"/>
            <xs:enumeration value="Unicode255"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionBasicTypeEnum">
        <xs:restriction base="xs:Name">
            <xs:pattern value="\p{Lu}[\p{L},\d]*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tBasicTypeEnum">
        <xs:union memberTypes="tPredefinedBasicTypeEnum tExtensionBasicTypeEnum"/>
    </xs:simpleType>
    <xs:simpleType name="tValKindEnum">
        <xs:restriction base="xs:Name">
            <xs:enumeration value="Spec"/>
            <xs:enumeration value="Conf"/>
            <xs:enumeration value="RO"/>
            <xs:enumeration value="Set"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tGSEControlTypeEnum">
        <xs:restriction base="xs:Name">
```

```
            <xs:enumeration value="GSSE"/>
            <xs:enumeration value="GOOSE"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tSIUnitEnum">
        <xs:restriction base="xs:token">
            <xs:enumeration value="none"/>
            <xs:enumeration value="m"/>
            <xs:enumeration value="kg"/>
            <xs:enumeration value="s"/>
            <xs:enumeration value="A"/>
            <xs:enumeration value="K"/>
            <xs:enumeration value="mol"/>
            <xs:enumeration value="cd"/>
            <xs:enumeration value="deg"/>
            <xs:enumeration value="rad"/>
            <xs:enumeration value="sr"/>
            <xs:enumeration value="Gy"/>
            <xs:enumeration value="q"/>
            <xs:enumeration value="°C"/>
            <xs:enumeration value="Sv"/>
            <xs:enumeration value="F"/>
            <xs:enumeration value="C"/>
            <xs:enumeration value="S"/>
            <xs:enumeration value="H"/>
            <xs:enumeration value="V"/>
            <xs:enumeration value="ohm"/>
            <xs:enumeration value="J"/>
            <xs:enumeration value="N"/>
            <xs:enumeration value="Hz"/>
            <xs:enumeration value="lx"/>
            <xs:enumeration value="Lm"/>
            <xs:enumeration value="Wb"/>
            <xs:enumeration value="T"/>
            <xs:enumeration value="W"/>
            <xs:enumeration value="Pa"/>
            <xs:enumeration value="m^2"/>
            <xs:enumeration value="m^3"/>
            <xs:enumeration value="m/s"/>
            <xs:enumeration value="m/s^2"/>
            <xs:enumeration value="m^3/s"/>
            <xs:enumeration value="m/m^3"/>
            <xs:enumeration value="M"/>
            <xs:enumeration value="kg/m^3"/>
            <xs:enumeration value="m^2/s"/>
            <xs:enumeration value="W/m K"/>
            <xs:enumeration value="J/K"/>
            <xs:enumeration value="ppm"/>
            <xs:enumeration value="s^-1"/>
            <xs:enumeration value="rad/s"/>
            <xs:enumeration value="VA"/>
            <xs:enumeration value="VAr"/>
            <xs:enumeration value="theta"/>
            <xs:enumeration value="cos_theta"/>
            <xs:enumeration value="Vs"/>
            <xs:enumeration value="V^2"/>
            <xs:enumeration value="As"/>
            <xs:enumeration value="A^2"/>
            <xs:enumeration value="A^2 s"/>
            <xs:enumeration value="VAh"/>
            <xs:enumeration value="Wh"/>
            <xs:enumeration value="VArh"/>
            <xs:enumeration value="V/Hz"/>
            <xs:enumeration value="b/s"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tUnitMultiplierEnum">
        <xs:restriction base="xs:normalizedString">
            <xs:enumeration value=""/>
            <xs:enumeration value="m"/>
            <xs:enumeration value="k"/>
            <xs:enumeration value="M"/>
```

```
            <xs:enumeration value="mu"/>
            <xs:enumeration value="y"/>
            <xs:enumeration value="z"/>
            <xs:enumeration value="a"/>
            <xs:enumeration value="f"/>
            <xs:enumeration value="p"/>
            <xs:enumeration value="n"/>
            <xs:enumeration value="c"/>
            <xs:enumeration value="d"/>
            <xs:enumeration value="da"/>
            <xs:enumeration value="h"/>
            <xs:enumeration value="G"/>
            <xs:enumeration value="T"/>
            <xs:enumeration value="P"/>
            <xs:enumeration value="E"/>
            <xs:enumeration value="Z"/>
            <xs:enumeration value="Y"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

File **SCL_BaseTypes.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_Enums.xsd"/>
    <xs:attributeGroup name="agDesc">
        <xs:attribute name="desc" type="xs:normalizedString" use="optional"/>
    </xs:attributeGroup>
    <xs:complexType name="tBaseElement" abstract="true">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Text" type="tText" minOccurs="0"/>
            <xs:element name="Private" type="tPrivate" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="tUnNaming" abstract="true">
        <xs:complexContent>
            <xs:extension base="tBaseElement">
                <xs:attributeGroup ref="agDesc"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tNaming" abstract="true">
        <xs:complexContent>
            <xs:extension base="tBaseElement">
                <xs:attribute name="name" type="tName" use="required"/>
                <xs:attributeGroup ref="agDesc"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tIDNaming" abstract="true">
        <xs:complexContent>
            <xs:extension base="tBaseElement">
                <xs:attribute name="id" type="tName" use="required"/>
                <xs:attributeGroup ref="agDesc"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tAnyContentFromOtherNamespace" abstract="true" mixed="true">
        <xs:annotation>
```

```
            <xs:documentation xml:lang="en">An element of this type can contain text mixed with elements from another namespace that this target namespace (but they must be defined in a namespace). Attributes from other namespaces than this target namespace are also allowed.</xs:documentation>
        </xs:annotation>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:any namespace="##other" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <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>
    <xs:complexType name="tPrivate" 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 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>
    <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"/>
        <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>
    <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>
    <xs:complexType name="tVal">
        <xs:simpleContent>
            <xs:extension base="xs:normalizedString">
                <xs:attribute name="sGroup" type="xs:unsignedInt" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tValueWithUnit">
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required"/>
                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tVoltage">
        <xs:simpleContent>
            <xs:restriction base="tValueWithUnit">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="V"/>
                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tBitRateInMbPerSec">
        <xs:simpleContent>
            <xs:restriction base="tValueWithUnit">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="b/s"/>
                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" fixed="M"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tDurationInSec">
        <xs:simpleContent>
            <xs:restriction base="tValueWithUnit">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="s"/>
                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" use="optional"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tDurationInMilliSec">
        <xs:simpleContent>
            <xs:restriction base="tValueWithUnit">
                <xs:attribute name="unit" type="tSIUnitEnum" use="required" fixed="s"/>
                <xs:attribute name="multiplier" type="tUnitMultiplierEnum" fixed="m"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

A.2 Substation syntax

File **SCL_Substation. xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <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>
    <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>
    <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>
    <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>
    <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>
    <xs:complexType name="tLNode">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:attribute name="lnInst" type="tAnyName" use="optional"/>
                <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"/>
                <xs:attribute name="prefix" type="tAnyName" use="optional"/>
                <xs:attribute name="lnType" type="tName" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <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>
    <xs:complexType name="tConnectivityNode">
        <xs:complexContent>
            <xs:extension base="tLNodeContainer">
                <xs:attribute name="pathName" type="tRef" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <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>
    <xs:element name="Substation" type="tSubstation">
        <xs:unique name="uniqueVoltageLevelInSubstation">
            <xs:selector xpath="./scl:VoltageLevel"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:unique name="uniquePowerTranformerInSubstation">
            <xs:selector xpath="./scl:PowerTransformer"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:unique name="uniqueGeneralEquipmentInSubstation">
            <xs:selector xpath="./scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:unique name="uniqueFunctionInSubstation">
            <xs:selector xpath="./scl:Function"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="ConnectivityNodeKey">
            <xs:selector xpath=".//scl:ConnectivityNode"/>
            <xs:field xpath="@pathName"/>
        </xs:key>
        <xs:unique name="uniqueLNode">
            <xs:selector xpath=".//scl:LNode"/>
            <xs:field xpath="@lnInst"/>
            <xs:field xpath="@lnClass"/>
            <xs:field xpath="@iedName"/>
            <xs:field xpath="@ldInst"/>
            <xs:field xpath="@prefix"/>
        </xs:unique>
        <xs:unique name="uniqueChildNameInSubstation">
            <xs:selector xpath="./*"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <!-- This identity constraint must be removed, as there is a problem with the (according to part 6 text)
    predefined connectivity node grounded. If a terminal references this node, which
    is naturally NOT defined explicitely in the SCL file, verification fails.
    <xs:keyref name="ref2ConnectivityNode" refer="ConnectivityNodeKey">
      <xs:selector xpath=".//scl:Terminal"/>
      <xs:field xpath="@connectivityNode"/>
```

```
    </xs:keyref>
    -->
    </xs:element>
</xs:schema>
```

A.3 Data type templates

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns="http://www.iec.ch/61850/2003/SCL"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_BaseTypes.xsd"/>
    <xs:attributeGroup name="agDATrgOp">
        <xs:attribute name="dchg" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="qchg" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dupd" type="xs:boolean" use="optional" default="false"/>
    </xs:attributeGroup>
    <xs:complexType name="tAbstractDataAttribute" abstract="true">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="Val" type="tVal" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="name" type="tAttributeNameEnum" use="required"/>
                <xs:attribute name="sAddr" type="xs:normalizedString" use="optional"/>
                <xs:attribute name="bType" type="tBasicTypeEnum" use="required"/>
                <xs:attribute name="valKind" type="tValKindEnum" use="optional" default="Set"/>
                <xs:attribute name="type" type="tAnyName" use="optional"/>
                <xs:attribute name="count" type="xs:unsignedInt" use="optional" default="0"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tLNodeType">
        <xs:complexContent>
            <xs:extension base="tIDNaming">
                <xs:sequence>
                    <xs:element name="DO" type="tDO" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="iedType" type="tAnyName" use="optional"/>
                <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDO">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:attribute name="name" type="tRestrName1stU" use="required"/>
                <xs:attribute name="type" type="tName" use="required"/>
                <xs:attribute name="accessControl" type="xs:normalizedString" use="optional"/>
                <xs:attribute name="transient" type="xs:boolean" use="optional" default="false"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDOType">
        <xs:complexContent>
            <xs:extension base="tIDNaming">
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="SDO" type="tSDO"/>
                    <xs:element name="DA" type="tDA"/>
                </xs:choice>
                <xs:attribute name="iedType" type="tAnyName" use="optional"/>
                <xs:attribute name="cdc" type="tCDCEnum" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSDO">
        <xs:complexContent>
            <xs:extension base="tNaming">
                <xs:attribute name="type" type="tName" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

```
<xs:complexType name="tDA">
    <xs:complexContent>
        <xs:extension base="tAbstractDataAttribute">
            <xs:attributeGroup ref="agDATrgOp"/>
            <xs:attribute name="fc" type="tFCEnum" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tDAType">
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:sequence>
                <xs:element name="BDA" type="tBDA" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="iedType" type="tAnyName" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tBDA">
    <xs:complexContent>
        <xs:extension base="tAbstractDataAttribute"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tEnumType">
    <xs:complexContent>
        <xs:extension base="tIDNaming">
            <xs:sequence>
                <xs:element name="EnumVal" type="tEnumVal" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tEnumVal">
    <xs:simpleContent>
        <xs:extension base="xs:normalizedString">
            <xs:attribute name="ord" type="xs:integer" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tDataTypeTemplates">
    <xs:sequence>
        <xs:element name="LNodeType" type="tLNodeType" maxOccurs="unbounded">
            <xs:unique name="uniqueDOInLNodeType">
                <xs:selector xpath="scl:DO"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="DOType" type="tDOType" maxOccurs="unbounded">
            <xs:unique name="uniqueDAorSDOInLDOType">
                <xs:selector xpath="./*"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="DAType" type="tDAType" minOccurs="0" maxOccurs="unbounded">
            <xs:unique name="uniqueBDAInLDAType">
                <xs:selector xpath="scl:BDA"/>
                <xs:field xpath="@name"/>
            </xs:unique>
        </xs:element>
        <xs:element name="EnumType" type="tEnumType" minOccurs="0" maxOccurs="unbounded">
            <xs:unique name="uniqueOrdInEnumType">
                <xs:selector xpath="scl:EnumVal"/>
                <xs:field xpath="@ord"/>
            </xs:unique>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:element name="DataTypeTemplates" type="tDataTypeTemplates">
    <xs:unique name="uniqueLNodeType">
        <xs:selector xpath="scl:LNodeType"/>
        <xs:field xpath="@id"/>
        <xs:field xpath="@iedType"/>
```

```
        </xs:unique>
        <xs:key name="DOTypeKey">
            <xs:selector xpath="scl:DOType"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:keyref name="ref2DOType" refer="DOTypeKey">
            <xs:selector xpath="scl:LNodeType/scl:DO"/>
            <xs:field xpath="@type"/>
        </xs:keyref>
        <xs:keyref name="ref2DOTypeForSDO" refer="DOTypeKey">
            <xs:selector xpath="scl:DOType/scl:SDO"/>
            <xs:field xpath="@type"/>
        </xs:keyref>
        <xs:key name="DATypeKey">
            <xs:selector xpath="scl:DAType"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="EnumTypeKey">
            <xs:selector xpath="scl:EnumType"/>
            <xs:field xpath="@id"/>
        </xs:key>
    </xs:element>
</xs:schema>
```

A.4 IED capabilities and structure

File **SCL_IED.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release
2003/09/19</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_BaseTypes.xsd"/>
    <xs:attributeGroup name="agAuthentication">
        <xs:attribute name="none" type="xs:boolean" use="optional" default="true"/>
        <xs:attribute name="password" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="weak" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="strong" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="certificate" type="xs:boolean" use="optional" default="false"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agSmvOpts">
        <xs:attribute name="refreshTime" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="sampleSynchronized" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="sampleRate" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="security" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dataRef" type="xs:boolean" use="optional" default="false"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agOptFields">
        <xs:attribute name="seqNum" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="timeStamp" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dataSet" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="reasonCode" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="dataRef" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="bufOvfl" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="entryID" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="configRef" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="segmentation" type="xs:boolean" use="optional" default="false"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agLDRef">
        <xs:attribute name="iedName" type="tName" use="required"/>
        <xs:attribute name="ldInst" type="tName" use="required"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agLNRef">
        <xs:attributeGroup ref="agLDRef"/>
        <xs:attribute name="prefix" type="tAnyName" use="optional"/>
        <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
        <xs:attribute name="lnInst" type="tAnyName" use="required"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agDORef">
        <xs:attributeGroup ref="agLNRef"/>
        <xs:attribute name="doName" type="tName" use="required"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="agDARef">
        <xs:attributeGroup ref="agDORef"/>
        <xs:attribute name="daName" type="tName" use="optional"/>
        <xs:attribute name="fc" type="tFCEnum" use="optional"/>
    </xs:attributeGroup>
    <xs:complexType name="tIED">
        <xs:complexContent>
            <xs:extension base="tNaming">
                <xs:sequence>
                    <xs:element name="Services" type="tServices" minOccurs="0"/>
                    <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
                        <xs:unique name="uniqueLNInAccessPoint">
                            <xs:selector xpath="./scl:LN"/>
                            <xs:field xpath="@inst"/>
                            <xs:field xpath="@lnClass"/>
                            <xs:field xpath="@prefix"/>
                        </xs:unique>
                    </xs:element>
```

```
            </xs:sequence>
            <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="manufacturer" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="configVersion" type="xs:normalizedString" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tServices">
    <xs:all>
        <xs:element name="DynAssociation" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="SettingGroups" minOccurs="0">
            <xs:complexType>
                <xs:all>
                    <xs:element name="SGEdit" type="tServiceYesNo" minOccurs="0"/>
                    <xs:element name="ConfSG" type="tServiceYesNo" minOccurs="0"/>
                </xs:all>
            </xs:complexType>
        </xs:element>
        <xs:element name="GetDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="GetDataObjectDefinition" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="DataObjectDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="GetDataSetValue" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="SetDataSetValue" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="DataSetDirectory" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfDataSet" type="tServiceWithMaxAndMaxAttributes" minOccurs="0"/>
        <xs:element name="DynDataSet" type="tServiceWithMaxAndMaxAttributes" minOccurs="0"/>
        <xs:element name="ReadWrite" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="TimerActivatedControl" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfReportControl" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="GetCBValues" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfLogControl" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="ReportSettings" type="tReportSettings" minOccurs="0"/>
        <xs:element name="LogSettings" type="tLogSettings" minOccurs="0"/>
        <xs:element name="GSESettings" type="tGSESettings" minOccurs="0"/>
        <xs:element name="SMVSettings" type="tSMVSettings" minOccurs="0"/>
        <xs:element name="GSEDir" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="GOOSE" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="GSSE" type="tServiceWithMax" minOccurs="0"/>
        <xs:element name="FileHandling" type="tServiceYesNo" minOccurs="0"/>
        <xs:element name="ConfLNs" type="tConfLNs" minOccurs="0"/>
    </xs:all>
</xs:complexType>
<xs:complexType name="tAccessPoint">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:choice minOccurs="0">
                <xs:element name="Server" type="tServer">
                    <xs:unique name="uniqueAssociationInServer">
                        <xs:selector xpath="./scl:Association"/>
                        <xs:field xpath="@associationID"/>
                    </xs:unique>
                </xs:element>
                <xs:element ref="LN" maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute name="router" type="xs:boolean" use="optional" default="false"/>
            <xs:attribute name="clock" type="xs:boolean" use="optional" default="false"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tServer">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Authentication">
                    <xs:complexType>
                        <xs:attributeGroup ref="agAuthentication"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="LDevice" type="tLDevice" maxOccurs="unbounded">
                    <xs:unique name="uniqueLNInLDevice">
                        <xs:selector xpath="./scl:LN"/>
                        <xs:field xpath="@inst"/>
```

```
                        <xs:field xpath="@lnClass"/>
                        <xs:field xpath="@prefix"/>
                    </xs:unique>
                </xs:element>
                <xs:element name="Association" type="tAssociation" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="timeout" type="xs:unsignedInt" use="optional" default="30"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tLDevice">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element ref="LN0"/>
                <xs:element ref="LN" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="AccessControl" type="tAccessControl" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="inst" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAccessControl" mixed="true">
    <xs:complexContent mixed="true">
        <xs:extension base="tAnyContentFromOtherNamespace"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAssociation">
    <xs:attribute name="kind" type="tAssociationKindEnum" use="required"/>
    <xs:attribute name="associationID" type="tName" use="optional"/>
    <xs:attributeGroup ref="agLNRef"/>
</xs:complexType>
<xs:element name="LN0">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="tLN0"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:unique name="uniqueReportControlInLN0">
        <xs:selector xpath="./scl:ReportControl"/>
        <xs:field xpath="@name"/>
    </xs:unique>
    <xs:unique name="uniqueLogControlInLN0">
        <xs:selector xpath="./scl:LogControl"/>
        <xs:field xpath="@name"/>
    </xs:unique>
    <xs:unique name="uniqueGSEControlInLN0">
        <xs:selector xpath="./scl:GSEControl"/>
        <xs:field xpath="@name"/>
    </xs:unique>
    <xs:unique name="uniqueSampledValueControlInLN0">
        <xs:selector xpath="./scl:SampledValueControl"/>
        <xs:field xpath="@name"/>
    </xs:unique>
    <xs:key name="DataSetKeyLN0">
        <xs:selector xpath="./scl:DataSet"/>
        <xs:field xpath="@name"/>
    </xs:key>
    <xs:keyref name="ref2DataSetReportLN0" refer="DataSetKeyLN0">
        <xs:selector xpath="./scl:ReportControl"/>
        <xs:field xpath="@datSet"/>
    </xs:keyref>
    <xs:keyref name="ref2DataSetLogLN0" refer="DataSetKeyLN0">
        <xs:selector xpath="./scl:LogControl"/>
        <xs:field xpath="@datSet"/>
    </xs:keyref>
    <xs:keyref name="ref2DataSetGSELN0" refer="DataSetKeyLN0">
        <xs:selector xpath="./scl:GSEControl"/>
        <xs:field xpath="@datSet"/>
    </xs:keyref>
    <xs:keyref name="ref2DataSetSVLN0" refer="DataSetKeyLN0">
        <xs:selector xpath="./scl:SampledValueControl"/>
```

```
            <xs:field xpath="@datSet"/>
        </xs:keyref>
    </xs:element>
    <xs:element name="LN" type="tLN">
        <xs:unique name="uniqueReportControlInLN">
            <xs:selector xpath="./scl:ReportControl"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:unique name="uniqueLogControlInLN">
            <xs:selector xpath="./scl:LogControl"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="DataSetKeyInLN">
            <xs:selector xpath="./scl:DataSet"/>
            <xs:field xpath="@name"/>
        </xs:key>
        <xs:keyref name="ref2DataSetReport" refer="DataSetKeyInLN">
            <xs:selector xpath="./scl:ReportControl"/>
            <xs:field xpath="@datSet"/>
        </xs:keyref>
        <xs:keyref name="ref2DataSetLog" refer="DataSetKeyInLN">
            <xs:selector xpath="./scl:LogControl"/>
            <xs:field xpath="@datSet"/>
        </xs:keyref>
    </xs:element>
    <xs:complexType name="tAnyLN" abstract="true">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="DataSet" type="tDataSet" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="ReportControl" type="tReportControl" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="LogControl" type="tLogControl" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="DOI" type="tDOI" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="Inputs" type="tInputs" minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="lnType" type="tName" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tLN">
        <xs:complexContent>
            <xs:extension base="tAnyLN">
                <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
                <xs:attribute name="inst" type="xs:unsignedInt" use="required"/>
                <xs:attribute name="prefix" type="tAnyName" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tLN0">
        <xs:complexContent>
            <xs:extension base="tAnyLN">
                <xs:sequence>
                    <xs:element name="GSEControl" type="tGSEControl" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="SampledValueControl" type="tSampledValueControl" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="SettingControl" type="tSettingControl" minOccurs="0"/>
                    <xs:element name="SCLControl" type="tSCLControl" minOccurs="0"/>
                    <xs:element name="Log" type="tLog" minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="lnClass" type="tLNClassEnum" use="required" fixed="LLN0"/>
                <xs:attribute name="inst" type="xs:normalizedString" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDataSet">
        <xs:complexContent>
            <xs:extension base="tNaming">
                <xs:sequence>
                    <xs:element name="FCDA" type="tFCDA" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
```

```
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tFCDA">
    <xs:attribute name="ldInst" type="tName" use="optional"/>
    <xs:attribute name="prefix" type="tAnyName" use="optional"/>
    <xs:attribute name="lnClass" type="tLNClassEnum" use="optional"/>
    <xs:attribute name="lnInst" type="tName" use="optional"/>
    <xs:attribute name="doName" type="tName" use="optional"/>
    <xs:attribute name="daName" type="tName" use="optional"/>
    <xs:attribute name="fc" type="tFCEnum" use="required"/>
</xs:complexType>
<xs:complexType name="tControl" abstract="true">
    <xs:complexContent>
        <xs:extension base="tNaming">
            <xs:attribute name="datSet" type="tAnyName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tControlWithTriggerOpt" abstract="true">
    <xs:complexContent>
        <xs:extension base="tControl">
            <xs:sequence>
                <xs:element name="TrgOps" type="tTrgOps" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="intgPd" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tTrgOps">
    <xs:attribute name="dchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="qchg" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dupd" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="period" type="xs:boolean" use="optional" default="false"/>
</xs:complexType>
<xs:complexType name="tReportControl">
    <xs:complexContent>
        <xs:extension base="tControlWithTriggerOpt">
            <xs:sequence>
                <xs:element name="OptFields">
                    <xs:complexType>
                        <xs:attributeGroup ref="agOptFields"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="RptEnabled" type="tRptEnabled" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="rptID" type="tName" use="required"/>
            <xs:attribute name="confRev" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="buffered" type="xs:boolean" use="optional" default="false"/>
            <xs:attribute name="bufTime" type="xs:unsignedInt" use="optional" default="0"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tRptEnabled">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="ClientLN" type="tClientLN" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="max" type="xs:unsignedInt" use="optional" default="1"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tClientLN">
    <xs:attributeGroup ref="agLNRef"/>
</xs:complexType>
<xs:complexType name="tLogControl">
    <xs:complexContent>
        <xs:extension base="tControlWithTriggerOpt">
            <xs:attribute name="logName" type="tName" use="required"/>
            <xs:attribute name="logEna" type="xs:boolean" use="optional" default="true"/>
            <xs:attribute name="reasonCode" type="xs:boolean" use="optional" default="true"/>
        </xs:extension>
```

```
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tInputs">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="ExtRef" type="tExtRef" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tExtRef">
        <xs:attributeGroup ref="agDORef"/>
        <xs:attribute name="daName" type="tName" use="optional"/>
        <xs:attribute name="intAddr" type="xs:normalizedString" use="optional"/>
    </xs:complexType>
    <xs:complexType name="tLog" mixed="true">
        <xs:complexContent mixed="true">
            <xs:extension base="tAnyContentFromOtherNamespace"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tControlWithIEDName">
        <xs:complexContent>
            <xs:extension base="tControl">
                <xs:sequence>
                    <xs:element name="IEDName" type="tName" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="confRev" type="xs:unsignedInt" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tGSEControl">
        <xs:complexContent>
            <xs:extension base="tControlWithIEDName">
                <xs:attribute name="type" type="tGSEControlTypeEnum" use="optional" default="GOOSE"/>
                <xs:attribute name="appID" type="xs:normalizedString" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSampledValueControl">
        <xs:complexContent>
            <xs:extension base="tControlWithIEDName">
                <xs:sequence>
                    <xs:element name="SmvOpts">
                        <xs:complexType>
                            <xs:attributeGroup ref="agSmvOpts"/>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="smvID" type="xs:normalizedString" use="required"/>
                <xs:attribute name="multicast" type="xs:boolean" default="true"/>
                <xs:attribute name="smpRate" type="xs:unsignedInt" use="required"/>
                <xs:attribute name="nofASDU" type="xs:unsignedInt" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSettingControl">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:attribute name="numOfSGs" type="xs:unsignedInt" use="required"/>
                <xs:attribute name="actSG" type="xs:unsignedInt" use="optional" default="1"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSCLControl">
        <xs:complexContent>
            <xs:extension base="tUnNaming"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDOI">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
```

```
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="SDI" type="tSDI"/>
                <xs:element name="DAI" type="tDAI"/>
            </xs:choice>
            <xs:attribute name="name" type="tRestrName1stU" use="required"/>
            <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
            <xs:attribute name="accessControl" type="xs:normalizedString" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSDI">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="SDI" type="tSDI"/>
                <xs:element name="DAI" type="tDAI"/>
            </xs:choice>
            <xs:attribute name="name" type="tRestrName1stL" use="required"/>
            <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tDAI">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:sequence>
                <xs:element name="Val" type="tVal" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" type="tRestrName1stL" use="required"/>
            <xs:attribute name="sAddr" type="xs:normalizedString" use="optional"/>
            <xs:attribute name="valKind" type="tValKindEnum" use="optional" default="Set"/>
            <xs:attribute name="ix" type="xs:unsignedInt" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tServiceYesNo"/>
<xs:complexType name="tServiceWithMax">
    <xs:attribute name="max" type="xs:unsignedInt" use="required"/>
</xs:complexType>
<xs:complexType name="tServiceWithMaxAndMaxAttributes">
    <xs:complexContent>
        <xs:extension base="tServiceWithMax">
            <xs:attribute name="maxAttributes" type="xs:unsignedInt" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tServiceWithMaxAndModify">
    <xs:complexContent>
        <xs:extension base="tServiceWithMax">
            <xs:attribute name="modify" type="xs:boolean" use="optional" default="true"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tServiceSettings" abstract="true">
    <xs:attribute name="cbName" type="tServiceSettingsEnum" use="optional" default="Fix"/>
    <xs:attribute name="datSet" type="tServiceSettingsEnum" use="optional" default="Fix"/>
</xs:complexType>
<xs:complexType name="tReportSettings">
    <xs:complexContent>
        <xs:extension base="tServiceSettings">
            <xs:attribute name="rptID" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            <xs:attribute name="optFields" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            <xs:attribute name="bufTime" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            <xs:attribute name="trgOps" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            <xs:attribute name="intgPd" type="tServiceSettingsEnum" use="optional" default="Fix"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tLogSettings">
    <xs:complexContent>
        <xs:extension base="tServiceSettings">
            <xs:attribute name="logEna" type="tServiceSettingsEnum" use="optional" default="Fix"/>
```

```
                <xs:attribute name="trgOps" type="tServiceSettingsEnum" use="optional" default="Fix"/>
                <xs:attribute name="intgPd" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tGSESettings">
        <xs:complexContent>
            <xs:extension base="tServiceSettings">
                <xs:attribute name="appID" type="tServiceSettingsEnum" use="optional" default="Fix"/>
                <xs:attribute name="dataLabel" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSMVSettings">
        <xs:complexContent>
            <xs:extension base="tServiceSettings">
                <xs:sequence>
                    <xs:element name="SmpRate" maxOccurs="unbounded">
                        <xs:simpleType>
                            <xs:restriction base="xs:decimal">
                                <xs:minInclusive value="0"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="svID" type="tServiceSettingsEnum" use="optional" default="Fix"/>
                <xs:attribute name="optFields" type="tServiceSettingsEnum" use="optional" default="Fix"/>
                <xs:attribute name="smpRate" type="tServiceSettingsEnum" use="optional" default="Fix"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tConfLNs">
        <xs:attribute name="fixPrefix" type="xs:boolean" use="optional" default="false"/>
        <xs:attribute name="fixLnInst" type="xs:boolean" use="optional" default="false"/>
    </xs:complexType>
    <xs:element name="IED" type="tIED">
        <xs:unique name="uniqueAccessPointInIED">
            <xs:selector xpath="./scl:AccessPoint"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="LDeviceInIEDKey">
            <xs:selector xpath="./scl:AccessPoint/scl:Server/scl:LDevice"/>
            <xs:field xpath="@inst"/>
        </xs:key>
        <xs:keyref name="ref2LDeviceInIED" refer="LDeviceInIEDKey">
            <xs:selector xpath="./scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:LogControl"/>
            <xs:field xpath="@logName"/>
        </xs:keyref>
    </xs:element>
</xs:schema>
```

A.5 Communication subnetworks

File **SCL_Communication.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns="http://www.iec.ch/61850/2003/SCL"
xmlns:scl="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_BaseTypes.xsd"/>
    <xs:complexType name="tControlBlock" abstract="true">
        <xs:annotation>
            <xs:documentation xml:lang="en">A control block within a Logical Device.</xs:documentation>
        </xs:annotation>
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="Address" type="tAddress" minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="ldInst" type="tName" use="required"/>
                <xs:attribute name="cbName" type="tName" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tCommunication">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
                <xs:sequence>
                    <xs:element name="SubNetwork" type="tSubNetwork" maxOccurs="unbounded">
                        <xs:unique name="uniqueConnectedAP">
                            <xs:selector xpath="./scl:ConnectedAP"/>
                            <xs:field xpath="@iedName"/>
                            <xs:field xpath="@apName"/>
                        </xs:unique>
                    </xs:element>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tSubNetwork">
        <xs:complexContent>
            <xs:extension base="tNaming">
                <xs:sequence>
                    <xs:element name="BitRate" type="tBitRateInMbPerSec" minOccurs="0"/>
                    <xs:element name="ConnectedAP" type="tConnectedAP" maxOccurs="unbounded">
                        <xs:unique name="uniqueGSEinConnectedAP">
                            <xs:selector xpath="./scl:GSE"/>
                            <xs:field xpath="@cbName"/>
                        </xs:unique>
                        <xs:unique name="uniqueSMVinConnectedAP">
                            <xs:selector xpath="./scl:SMV"/>
                            <xs:field xpath="@cbName"/>
                        </xs:unique>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="type" type="xs:normalizedString" use="optional">
                    <xs:annotation>
                        <xs:documentation xml:lang="en">The bus protocol types are defined in IEC 61850 Part 8 and
9</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tConnectedAP">
        <xs:complexContent>
            <xs:extension base="tUnNaming">
```

```
            <xs:sequence>
                <xs:element name="Address" type="tAddress" minOccurs="0"/>
                <xs:element name="GSE" type="tGSE" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="SMV" type="tSMV" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="PhysConn" type="tPhysConn" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="iedName" type="tName" use="required"/>
            <xs:attribute name="apName" type="tName" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAddress">
    <xs:sequence>
        <xs:element name="P" type="tP" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="tGSE">
    <xs:complexContent>
        <xs:extension base="tControlBlock">
            <xs:sequence>
                <xs:element name="MinTime" type="tDurationInMilliSec" minOccurs="0"/>
                <xs:element name="MaxTime" type="tDurationInMilliSec" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSMV">
    <xs:complexContent>
        <xs:extension base="tControlBlock"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tPhysConn">
    <xs:sequence>
        <xs:element name="P" type="tP" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:normalizedString" use="required"/>
</xs:complexType>
<xs:complexType name="tP">
    <xs:simpleContent>
        <xs:extension base="tPAddr">
            <xs:attribute name="type" type="tPTypeEnum" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_IP">
    <xs:annotation>
        <xs:documentation xml:lang="en">A TCP/IP address</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:pattern value="[0-2]?\d{1,2}\.[0-2]?\d{1,2}\.[0-2]?\d{1,2}.[0-2]?\d{1,2}"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="IP"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_IP-SUBNET">
    <xs:annotation>
        <xs:documentation xml:lang="en">A subnet Mask for TCP/IP profiles</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:pattern value="[0-2]?\d{1,2}\.[0-2]?\d{1,2}\.[0-2]?\d{1,2}.[0-2]?\d{1,2}"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="IP-SUBNET"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_IP-GATEWAY">
    <xs:annotation>
        <xs:documentation xml:lang="en">A First Hop IP gateway address for TCP/IP profiles</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
```

```
            <xs:pattern value="[0-2]?\d{1,2}\.[0-2]?\d{1,2}\.[0-2]?\d{1,2}.[0-2]?\d{1,2}"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="IP-GATEWAY"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-NSAP">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI Network Address</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="40"/>
            <xs:pattern value="[\d,A-F]+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-NSAP"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-TSEL">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI Transport Selector</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="8"/>
            <xs:pattern value="[\d,A-F]+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-TSEL"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-SSEL">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI Session Selector</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="16"/>
            <xs:pattern value="[\d,A-F]+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-SSEL"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-PSEL">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI Presentation Selector</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="16"/>
            <xs:pattern value="[\d,A-F]+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-PSEL"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-AP-Title">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI ACSE AP Title value</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:pattern value="&#34;[\d,&#44;]+&#34;"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-AP-Title"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-AP-Invoke">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI ACSE AP Invoke ID</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="5"/>
            <xs:pattern value="\d+"/>
```

```
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-AP-Invoke"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-AE-Qualifier">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI ACSE AE Qualifier</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="5"/>
            <xs:pattern value="\d+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-AE-Qualifier"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_OSI-AE-Invoke">
    <xs:annotation>
        <xs:documentation xml:lang="en">An OSI ACSE AE Invoke ID</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:maxLength value="5"/>
            <xs:pattern value="\d+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="OSI-AE-Invoke"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_MAC-Address">
    <xs:annotation>
        <xs:documentation xml:lang="en">A Media Access Address value</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:minLength value="17"/>
            <xs:maxLength value="17"/>
            <xs:pattern value="[\d,A-F]{2}\-[\d,A-F]{2}\-[\d,A-F]{2}\-[\d,A-F]{2}\-[\d,A-F]{2}\-[\d,A-F]{2}"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="MAC-Address"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_APPID">
    <xs:annotation>
        <xs:documentation xml:lang="en">An Application Identifier</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:minLength value="4"/>
            <xs:maxLength value="4"/>
            <xs:pattern value="[\d,A-F]+"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="APPID"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_VLAN-PRIORITY">
    <xs:annotation>
        <xs:documentation xml:lang="en">A VLAN User Priority</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:pattern value="[0-7]"/>
            <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="VLAN-PRIORITY"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_VLAN-ID">
    <xs:annotation>
        <xs:documentation xml:lang="en">A VLAN ID</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:restriction base="tP">
            <xs:minLength value="3"/>
```

```
                <xs:maxLength value="3"/>
                <xs:pattern value="[\d,A-F]+"/>
                <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="VLAN-ID"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <xs:element name="Communication" type="tCommunication">
        <xs:unique name="uniqueSubNetwork">
            <xs:selector xpath="./scl:SubNetwork"/>
            <xs:field xpath="@name"/>
        </xs:unique>
    </xs:element>
</xs:schema>
```

A.6 Main SCL

File **SCL.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" finalDefault="extension" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
(Uncommented)</xs:documentation>
    </xs:annotation>
    <xs:include schemaLocation="SCL_Substation.xsd"/>
    <xs:include schemaLocation="SCL_IED.xsd"/>
    <xs:include schemaLocation="SCL_Communication.xsd"/>
    <xs:include schemaLocation="SCL_DataTypeTemplates.xsd"/>
    <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:element ref="IED" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element ref="DataTypeTemplates" minOccurs="0"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:unique name="uniqueSubstation">
            <xs:selector xpath="./scl:Substation"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="IEDKey">
            <xs:selector xpath="./scl:IED"/>
            <xs:field xpath="@name"/>
        </xs:key>
        <xs:key name="LNodeTypeKey">
            <xs:selector xpath="./scl:DataTypeTemplates/scl:LNodeType"/>
            <xs:field xpath="@id"/>
            <xs:field xpath="@lnClass"/>
        </xs:key>
        <xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
    </xs:element>
</xs:schema>
```

Annex B
(normative)

SCL enumerations according to IEC 61850-7-3 and IEC 61850-7-4

```
<?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="Normative Enumerations 2003" nameStructure="IEDName"/>
     <DataTypeTemplates>
          <LNodeType id="Dummy" lnClass="LLN0">
               <DO name="Mod" type="myMod"/>
          </LNodeType>
          <DOType id="myMod" cdc="INC">
               <DA name="ctlVal" fc="CO" bType="Enum" " type="Mod"/>
               <DA name="stVal" fc="ST" bType="Enum" dchg="true" type="Mod"/>
               <DA name="q" fc="ST" bType="Quality" dchg="true"/>
               <DA name="t" fc="ST" bType="Timestamp" dchg="true"/>
          </DOType>
          <EnumType id="ctlModel">
               <EnumVal ord="0">status-only</EnumVal>
               <EnumVal ord="1">direct-with-normal-security</EnumVal>
               <EnumVal ord="2">sbo-with-normal-security</EnumVal>
               <EnumVal ord="3">direct-with-enhanced-security</EnumVal>
               <EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
          </EnumType>
          <EnumType id="sboClass">
               <EnumVal ord="0">operate-once</EnumVal>
               <EnumVal ord="1">operate-many</EnumVal>
          </EnumType>
          <EnumType id="orCategory">
               <EnumVal ord="0">not-supported</EnumVal>
               <EnumVal ord="1">bay-control</EnumVal>
               <EnumVal ord="2">station-control</EnumVal>
               <EnumVal ord="3">remote-control</EnumVal>
               <EnumVal ord="4">automatic-bay</EnumVal>
               <EnumVal ord="5">automatic-station</EnumVal>
               <EnumVal ord="6">automatic-remote</EnumVal>
               <EnumVal ord="7">maintenance</EnumVal>
               <EnumVal ord="8">process</EnumVal>
          </EnumType>
          <EnumType id="dir">
               <EnumVal ord="0">unknown</EnumVal>
               <EnumVal ord="1">forward</EnumVal>
               <EnumVal ord="2">backward</EnumVal>
               <EnumVal ord="3">both</EnumVal>
          </EnumType>
          <EnumType id="sev">
               <EnumVal ord="0">Unknown</EnumVal>
               <EnumVal ord="1">critical</EnumVal>
               <EnumVal ord="2">major</EnumVal>
               <EnumVal ord="3">minor</EnumVal>
               <EnumVal ord="4">warning</EnumVal>
          </EnumType>
          <EnumType id="range">
               <EnumVal ord="0">normal</EnumVal>
               <EnumVal ord="1">high</EnumVal>
               <EnumVal ord="2">low</EnumVal>
               <EnumVal ord="3">high-high</EnumVal>
               <EnumVal ord="4">low-low</EnumVal>
          </EnumType>
          <EnumType id="angidCMV">
               <EnumVal ord="0">V</EnumVal>
               <EnumVal ord="1">A</EnumVal>
               <EnumVal ord="2">other</EnumVal>
          </EnumType>
          <EnumType id="angid">
               <EnumVal ord="0">Va</EnumVal>
               <EnumVal ord="1">Vb</EnumVal>
```

```
        <EnumVal ord="2">Vc</EnumVal>
        <EnumVal ord="3">Aa</EnumVal>
        <EnumVal ord="4">Ab</EnumVal>
        <EnumVal ord="5">Ac</EnumVal>
        <EnumVal ord="6">Vab</EnumVal>
        <EnumVal ord="7">Vbc</EnumVal>
        <EnumVal ord="8">Vca</EnumVal>
        <EnumVal ord="9">Aother</EnumVal>
        <EnumVal ord="10">Vother</EnumVal>
    </EnumType>
    <EnumType id="phsid">
        <EnumVal ord="0">A</EnumVal>
        <EnumVal ord="1">B</EnumVal>
        <EnumVal ord="2">C</EnumVal>
    </EnumType>
    <EnumType id="seqT">
        <EnumVal ord="0">pos-neg-zero</EnumVal>
        <EnumVal ord="1">dir-quad-zero</EnumVal>
    </EnumType>
    <EnumType id="hvid">
        <EnumVal ord="0">fundamental</EnumVal>
        <EnumVal ord="1">rms</EnumVal>
        <EnumVal ord="2">absolute</EnumVal>
    </EnumType>
    <EnumType id="setCharact">
        <EnumVal ord="0"/>
        <EnumVal ord="1">ANSI Extremly Inverse</EnumVal>
        <EnumVal ord="2">ANSI Very Inverse</EnumVal>
        <EnumVal ord="3">ANSI Normal Inverse</EnumVal>
        <EnumVal ord="4">ANSI Moderate Inverse</EnumVal>
        <EnumVal ord="5">ANSI Definite Time</EnumVal>
        <EnumVal ord="6">Long-Time Extremely Inverse</EnumVal>
        <EnumVal ord="7">Long-Time Very Inverse</EnumVal>
        <EnumVal ord="8">Long-Time Inverse</EnumVal>
        <EnumVal ord="9">IEC Normal Inverse</EnumVal>
        <EnumVal ord="10">IEC Very Inverse</EnumVal>
        <EnumVal ord="11">IEC Inverse</EnumVal>
        <EnumVal ord="12">IEC Extremely Inverse</EnumVal>
        <EnumVal ord="13">IEC Short-Time Inverse</EnumVal>
        <EnumVal ord="14">IEC Long-Time Inverse</EnumVal>
        <EnumVal ord="15">IEC Definite Time</EnumVal>
        <EnumVal ord="16">Reserved</EnumVal>
    </EnumType>
    <EnumType id="multiplier">
        <EnumVal ord="-24">y</EnumVal>
        <EnumVal ord="-21">z</EnumVal>
        <EnumVal ord="-18">a</EnumVal>
        <EnumVal ord="-15">f</EnumVal>
        <EnumVal ord="-12">p</EnumVal>
        <EnumVal ord="-9">n</EnumVal>
        <EnumVal ord="-6">μ</EnumVal>
        <EnumVal ord="-3">m</EnumVal>
        <EnumVal ord="-2">c</EnumVal>
        <EnumVal ord="-1">d</EnumVal>
        <EnumVal ord="0"/>
        <EnumVal ord="1">da</EnumVal>
        <EnumVal ord="2">h</EnumVal>
        <EnumVal ord="3">k</EnumVal>
        <EnumVal ord="6">M</EnumVal>
        <EnumVal ord="9">G</EnumVal>
        <EnumVal ord="12">T</EnumVal>
        <EnumVal ord="15">P</EnumVal>
        <EnumVal ord="18">E</EnumVal>
        <EnumVal ord="21">Z</EnumVal>
        <EnumVal ord="24">Y</EnumVal>
    </EnumType>
    <EnumType id="SIUnit">
        <EnumVal ord="1"/>
        <EnumVal ord="2">m</EnumVal>
        <EnumVal ord="3">kg</EnumVal>
        <EnumVal ord="4">s</EnumVal>
        <EnumVal ord="5">A</EnumVal>
```

```
        <EnumVal ord="6">K</EnumVal>
        <EnumVal ord="7">mol</EnumVal>
        <EnumVal ord="8">cd</EnumVal>
        <EnumVal ord="9">deg</EnumVal>
        <EnumVal ord="10">rad</EnumVal>
        <EnumVal ord="11">sr</EnumVal>
        <EnumVal ord="21">Gy</EnumVal>
        <EnumVal ord="22">q</EnumVal>
        <EnumVal ord="23">°C</EnumVal>
        <EnumVal ord="24">Sv</EnumVal>
        <EnumVal ord="25">F</EnumVal>
        <EnumVal ord="26">C</EnumVal>
        <EnumVal ord="27">S</EnumVal>
        <EnumVal ord="28">H</EnumVal>
        <EnumVal ord="29">V</EnumVal>
        <EnumVal ord="30">ohm</EnumVal>
        <EnumVal ord="31">J</EnumVal>
        <EnumVal ord="32">N</EnumVal>
        <EnumVal ord="33">Hz</EnumVal>
        <EnumVal ord="34">lx</EnumVal>
        <EnumVal ord="35">Lm</EnumVal>
        <EnumVal ord="36">Wb</EnumVal>
        <EnumVal ord="37">T</EnumVal>
        <EnumVal ord="38">W</EnumVal>
        <EnumVal ord="39">Pa</EnumVal>
        <EnumVal ord="41">m²</EnumVal>
        <EnumVal ord="42">m³</EnumVal>
        <EnumVal ord="43">m/s</EnumVal>
        <EnumVal ord="44">m/s²</EnumVal>
        <EnumVal ord="45">m³/s</EnumVal>
        <EnumVal ord="46">m/m³</EnumVal>
        <EnumVal ord="47">M</EnumVal>
        <EnumVal ord="48">kg/m³</EnumVal>
        <EnumVal ord="49">m²/s</EnumVal>
        <EnumVal ord="50">W/m K</EnumVal>
        <EnumVal ord="51">J/K</EnumVal>
        <EnumVal ord="52">ppm</EnumVal>
        <EnumVal ord="53">1/s</EnumVal>
        <EnumVal ord="54">rad/s</EnumVal>
        <EnumVal ord="61">VA</EnumVal>
        <EnumVal ord="62">W</EnumVal>
        <EnumVal ord="63">VAr</EnumVal>
        <EnumVal ord="64">theta</EnumVal>
        <EnumVal ord="65">cos(theta)</EnumVal>
        <EnumVal ord="66">Vs</EnumVal>
        <EnumVal ord="67">V²</EnumVal>
        <EnumVal ord="68">As</EnumVal>
        <EnumVal ord="69">A²</EnumVal>
        <EnumVal ord="70">A²t</EnumVal>
        <EnumVal ord="71">VAh</EnumVal>
        <EnumVal ord="72">Wh</EnumVal>
        <EnumVal ord="73">VArh</EnumVal>
        <EnumVal ord="74">V/Hz</EnumVal>
    </EnumType>
    <EnumType id="Dbpos">
        <EnumVal ord="0">intermediate</EnumVal>
        <EnumVal ord="1">off</EnumVal>
        <EnumVal ord="2">on</EnumVal>
        <EnumVal ord="3">bad</EnumVal>
    </EnumType>
    <EnumType id="Tcmd">
        <EnumVal ord="0">stop</EnumVal>
        <EnumVal ord="1">lower</EnumVal>
        <EnumVal ord="2">higher</EnumVal>
        <EnumVal ord="3">reserved</EnumVal>
    </EnumType>
    <EnumType id="Beh">
        <EnumVal ord="1">on</EnumVal>
        <EnumVal ord="2">blocked</EnumVal>
        <EnumVal ord="3">test</EnumVal>
        <EnumVal ord="4">test/blocked</EnumVal>
        <EnumVal ord="5">off</EnumVal>
```

```
</EnumType>
<EnumType id="Mod">
    <EnumVal ord="1">on</EnumVal>
    <EnumVal ord="2">blocked</EnumVal>
    <EnumVal ord="3">test</EnumVal>
    <EnumVal ord="4">test/blocked</EnumVal>
    <EnumVal ord="5">off</EnumVal>
</EnumType>
<EnumType id="Health">
    <EnumVal ord="1">Ok</EnumVal>
    <EnumVal ord="2">Warning</EnumVal>
    <EnumVal ord="3">Alarm</EnumVal>
</EnumType>
<EnumType id="CBOpCap">
    <EnumVal ord="1">None</EnumVal>
    <EnumVal ord="2">Open</EnumVal>
    <EnumVal ord="3">Close-Open</EnumVal>
    <EnumVal ord="4">Open-Close-Open</EnumVal>
    <EnumVal ord="5">Close-Open-Close-Open</EnumVal>
</EnumType>
<EnumType id="DirMod">
    <EnumVal ord="1">NonDirectional</EnumVal>
    <EnumVal ord="2">Forward</EnumVal>
    <EnumVal ord="3">Inverse</EnumVal>
</EnumType>
<EnumType id="FailMod">
    <EnumVal ord="1">Current</EnumVal>
    <EnumVal ord="2">Breaker Status</EnumVal>
    <EnumVal ord="3">Both current and breaker status</EnumVal>
    <EnumVal ord="4">Other</EnumVal>
</EnumType>
<EnumType id="FanCtl">
    <EnumVal ord="1">Inactive</EnumVal>
    <EnumVal ord="2">Stage 1</EnumVal>
    <EnumVal ord="3">Stage 2</EnumVal>
    <EnumVal ord="4">Stage 3</EnumVal>
</EnumType>
<EnumType id="GnSt">
    <EnumVal ord="1">Stopped</EnumVal>
    <EnumVal ord="2">Stopping</EnumVal>
    <EnumVal ord="3">Started</EnumVal>
    <EnumVal ord="4">Starting</EnumVal>
    <EnumVal ord="5">Disabled</EnumVal>
</EnumType>
<EnumType id="LevMod">
    <EnumVal ord="1">Positive or Rising</EnumVal>
    <EnumVal ord="2">Negative or Falling</EnumVal>
    <EnumVal ord="3">Both</EnumVal>
    <EnumVal ord="4">Other</EnumVal>
</EnumType>
<EnumType id="LivDeaMod">
    <EnumVal ord="1">Dead Line, Dead Bus</EnumVal>
    <EnumVal ord="2">Live Line, Dead Bus</EnumVal>
    <EnumVal ord="3">Dead Line, Live Bus</EnumVal>
    <EnumVal ord="4">Dead Line, Dead Bus OR Live Line, Dead Bus</EnumVal>
    <EnumVal ord="5">Dead Line, Dead Bus OR Dead Line, Live Bus</EnumVal>
    <EnumVal ord="6">Live Line, Dead Bus OR Dead Line, Live Bus</EnumVal>
    <EnumVal ord="7">Dead Line, Dead Bus OR Live Line, Dead Bus OR Dead Line, Live Bus</EnumVal>
</EnumType>
<EnumType id="PolQty">
    <EnumVal ord="1">None</EnumVal>
    <EnumVal ord="2">Zero Sequence Current</EnumVal>
    <EnumVal ord="3">Zero Sequence Voltage</EnumVal>
    <EnumVal ord="4">Negative Sequence Voltage</EnumVal>
</EnumType>
<EnumType id="POWCap">
    <EnumVal ord="1">None</EnumVal>
    <EnumVal ord="2">Close</EnumVal>
    <EnumVal ord="3">Open</EnumVal>
    <EnumVal ord="4">Close and Open</EnumVal>
</EnumType>
<EnumType id="OpMod">
```

```
      <EnumVal ord="1">Overwrite existing values</EnumVal>
      <EnumVal ord="2">Stop when full or saturated</EnumVal>
    </EnumType>
    <EnumType id="ReTrMod">
      <EnumVal ord="1">Off</EnumVal>
      <EnumVal ord="2">Without Check</EnumVal>
      <EnumVal ord="3">With Current Check</EnumVal>
      <EnumVal ord="4">With Breaker Status Check</EnumVal>
      <EnumVal ord="5">With Current and Breaker Status Check</EnumVal>
      <EnumVal ord="6">Other Checks</EnumVal>
    </EnumType>
    <EnumType id="RstMod">
      <EnumVal ord="1">None</EnumVal>
      <EnumVal ord="2">Harmonic2</EnumVal>
      <EnumVal ord="3">Harmonic5</EnumVal>
      <EnumVal ord="4">Harmonic2and5</EnumVal>
      <EnumVal ord="5">WaveformAnalysis</EnumVal>
      <EnumVal ord="6">WaveformAnalysisAndHarmonic2</EnumVal>
      <EnumVal ord="7">Other</EnumVal>
    </EnumType>
    <EnumType id="RvAMod">
      <EnumVal ord="1">Off</EnumVal>
      <EnumVal ord="2">On</EnumVal>
    </EnumType>
    <EnumType id="SchTyp">
      <EnumVal ord="1">None</EnumVal>
      <EnumVal ord="2">Intertrip</EnumVal>
      <EnumVal ord="3">Permissive Underreach</EnumVal>
      <EnumVal ord="4">Permissive Overreach</EnumVal>
      <EnumVal ord="5">Blocking</EnumVal>
    </EnumType>
    <EnumType id="ShOpCap">
      <EnumVal ord="1">None</EnumVal>
      <EnumVal ord="2">Open</EnumVal>
      <EnumVal ord="3">Close</EnumVal>
      <EnumVal ord="4">Open and Close</EnumVal>
    </EnumType>
    <EnumType id="SwOpCap">
      <EnumVal ord="1">None</EnumVal>
      <EnumVal ord="2">Open</EnumVal>
      <EnumVal ord="3">Close</EnumVal>
      <EnumVal ord="4">Open and Close</EnumVal>
    </EnumType>
    <EnumType id="SwTyp">
      <EnumVal ord="1">Load Break</EnumVal>
      <EnumVal ord="2">Disconnector</EnumVal>
      <EnumVal ord="3">Earthing Switch</EnumVal>
      <EnumVal ord="4">High Speed Earthing Switch</EnumVal>
    </EnumType>
    <EnumType id="TrgMod">
      <EnumVal ord="1">Internal</EnumVal>
      <EnumVal ord="2">External</EnumVal>
      <EnumVal ord="3">Both</EnumVal>
    </EnumType>
    <EnumType id="TrMod">
      <EnumVal ord="1">3 phase tripping</EnumVal>
      <EnumVal ord="2">1 or 3 phase tripping</EnumVal>
      <EnumVal ord="3">specific</EnumVal>
    </EnumType>
    <EnumType id="TypRsCrv">
      <EnumVal ord="1">None</EnumVal>
      <EnumVal ord="2">Definit Time Delayed Reset</EnumVal>
      <EnumVal ord="3">Inverse Reset</EnumVal>
    </EnumType>
    <EnumType id="UnBlkMod">
      <EnumVal ord="1">Off</EnumVal>
      <EnumVal ord="2">Permanent</EnumVal>
      <EnumVal ord="3">Time window</EnumVal>
    </EnumType>
    <EnumType id="WeiMod">
      <EnumVal ord="1">Off</EnumVal>
      <EnumVal ord="2">Operate</EnumVal>
```

```
            <EnumVal ord="3">Echo</EnumVal>
            <EnumVal ord="4">Echo and Operate</EnumVal>
        </EnumType>
    </DataTypeTemplates>
</SCL>
```

Annex C
(informative)

Syntax extension examples

C.1 Extension syntax for drawing layout coordinates

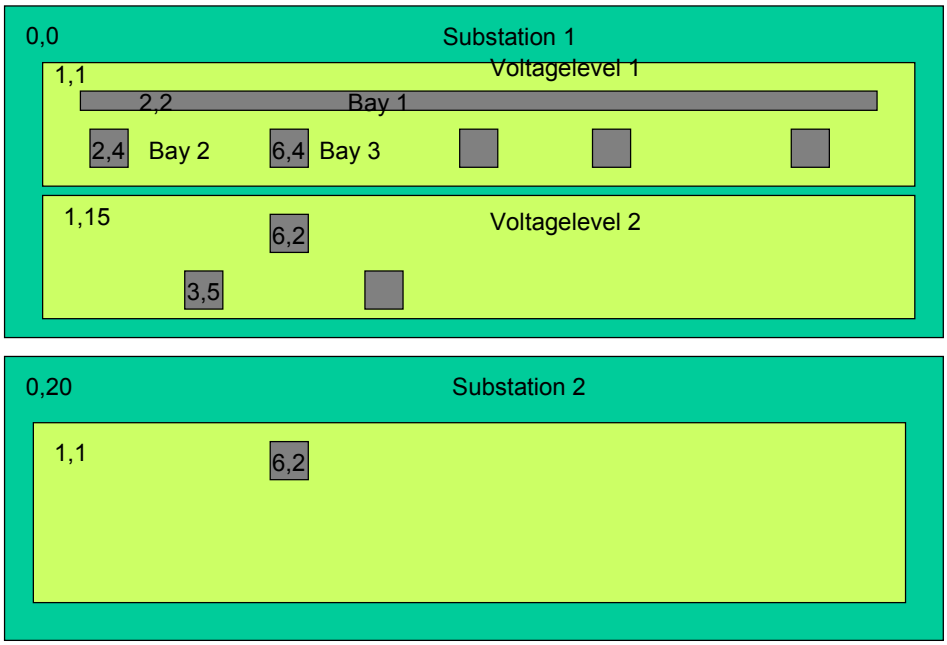
This annex defines a simple SCL extension to add coordinates to objects, so that they can be easily shown on a drawing. This is sufficient for a lot of drawing tasks, and serves here as an example of extension of the core SCL by another name space.

The handling (for example drawing) of object connections as well as the packaging of objects into drawing pages is private to the interpreting application. Typical drawings could be that of a substation as substation single line, a bay as bay single line, the communication section as a communication configuration drawing.

The coordinate system is a relative *x, y* system with coordinates using positive integer numbers. The point (0,0) is the upper left point of a drawing plane which is unlimited to downwards and right direction. The unit 1 principally refers to the size of an object. If different object sizes are used, then 1 is the size of the smallest object. However, transport of coordinates between different drawing applications might in this case lead to strange representations.

If coordinates are defined at different SCL tag hierarchy levels, then each level contains coordinates relative to the higher level. The absolute coordinate of a lower level is thus calculated by summing up all higher level coordinates, and the object coordinates themselves. If there are no coordinates defined at a higher level, then (x,y) = (0,0) is assumed.

This is illustrated in the Figure C.1. Here, for example, the bay 3 of substation 1 voltage level 1 has the absolute coordinates (0+1+6, 0+1+4) = (7,5) within a picture showing the substation 1, or even both substations.



IEC 212/04

Figure C.1 – Coordinate example

Aditional XML elements:

Only the additional XML attributes *x* and *y* for the coordinates in the x and the y direction are needed in addition to the SCL elements, which represent drawable objects. Additionally, the optional attribute *dir* with the value *horizontal* or *vertical* can optionally give the preferred connection direction of the object. If this attribute is defined at a bay, this means that all contained primary devices are oriented vertically, except those where another value of *dir* is explicitly stated. The coordinate name space shall be http://www.iec.ch/61850/2003/SCLcoordinates.

An appropriate XML schema definition is:

```
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCLcoordinates"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.iec.ch/61850/2003/SCLcoordinates"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.1">
    <xs:annotation>
        <xs:documentation xml:lang="en">
                    COPYRIGHT IEC, 2003. Version 1.0. Release 2003/09/19.
                    This schema is for infomational purposes only, and is not normative!
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType name="tConndir">
        <xs:restriction base="xs:normalizedString">
            <xs:enumeration value="horizontal"/>
            <xs:enumeration value="vertical"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:attribute name="x" type="xs:int"/>
    <xs:attribute name="y" type="xs:int"/>
    <xs:attribute name="dir" type="tConndir"/>
</xs:schema>
```

The following gives an SCL example using the coordinates. The transformer baden220_132.T1 in this example will have the coordinates (1,10) relative to the substation. The bay D1Q1 of voltage level D1 will be located in the upper left corner of the substation layout.

Observe that this is a standardized extension, therefore the extension name (sxy) does not start with an e. For private extensions, it shall start with an e (see 8.2.5).

```
<?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/SCLSCL.xsd"
xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates" sxy:schemaLocation="d:\data\IECTC57\SCL-
XML\SCLcoordinates.xsd">
    <Header id="SCL Example T1-1" nameStructure="IEDName"/>
    <Substation name="baden220_132" sxy:x="1" sxy:y="1" >
        <PowerTransformer name="T1" type="PTR" sxy:x="1" sxy:y="10" sxy:dir="horizontal">
            <TransformerWinding name="W1" type="PTW">
            </TransformerWinding>
            <TransformerWinding name="W2" type="PTW">
            </TransformerWinding>
        </PowerTransformer>
        <VoltageLevel name="D1" sxy:x="1" sxy:y="1">
            <Bay name="Q1" sxy:x="1" sxy:y="1" sxy:dir="horizontal"/>
        </VoltageLevel>
    </Substation>
</SCL>
```

C.2 Extension syntax for maintenance

This annex defines a simple SCL extension to indicate at LNodeType attributes, if they are mandatory, conditional, optional or private. As this is only needed for planning of system extensions, or by using the SCL syntax as general specification of CDCs, it is considered to be an extension package.

The name space for this extension package shall be *http://www.iec.ch/61850/2003/SCLmaintenance*. The namespace name shall be *smop* (xmlns:smop).

An appropriate XML schema definition is:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCLmaintenance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.iec.ch/61850/2003/SCLmaintenance"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.1">
    <xs:annotation>
        <xs:documentation xml:lang="en">COPYRIGHT IEC, 2003. Version 0.1. Draft 2003/08/28.</xs:documentation>
    </xs:annotation>
    <xs:simpleType name="tRestrName1stL">
        <xs:restriction base="xs:Name">
            <xs:pattern value="\p{Ll}[\d,\p{L},_]*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tMopEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="M"/>
            <xs:enumeration value="O"/>
            <xs:enumeration value="P"/>
            <xs:enumeration value="C"/>
            <xs:enumeration value="C1"/>
            <xs:enumeration value="C2"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tExtensionMopEnum">
        <xs:restriction base="tRestrName1stL"/>
    </xs:simpleType>
    <xs:simpleType name="tMOP">
        <xs:union memberTypes="tMopEnum tExtensionMopEnum"/>
    </xs:simpleType>
    <xs:attribute name="mop" type="tMOP"/>
    <xs:element name="CondDesc">
        <xs:complexType>
            <xs:attribute name="desc" type="xs:string" use="required"/>
            <xs:attribute ref="mop" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

This syntax defines an attribute mop with allowed values M, O, P and C, C1, C2. The mop value M means mandatory, O means optional, and P means private, i.e. specific to the manufacturer of the concerned IED type. Values C, C1, C2 are different conditions under which the relevant object is mandatory or not. More specific conditions for example as defined in IEC 61850-7-3 can be introduced additionally. Private conditions shall start with an E. For textual explanation of the conditions, the CondDesc element can be used.

NOTE The above syntax definition can not restrict the usage of the mop attribute. However, it shall only be used within the *DataTypeTemplate* section, at the elements DO, DA, SDO and BDA.

Annex D
(informative)

Example

D.1 Example specification

An example based on the specification in IEC 61850-5, I.1.3.2 is given here. The naming of devices is however changed to conform to the IEC 61346 series. Although this example is not 100 % complete, it illustrates most of the SCL possibilities for system description, i.e. it is a SCD file.

D.1.1 Substation configuration

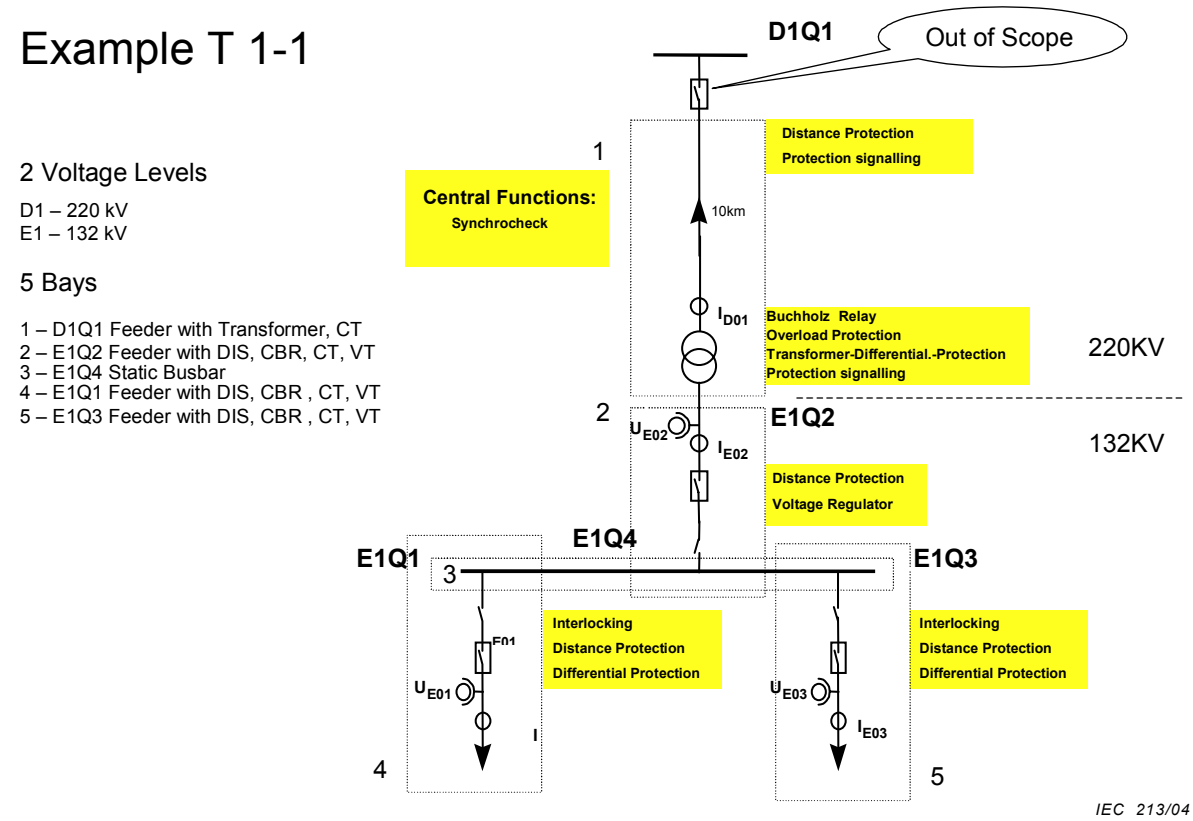


Figure D.1 – T1-1 Substation configuration

Figure D.1 shows the single line. The current infed via D1Q1 to the transformer D1T1 is distributed at the lower voltage side to two lines E1Q1 and E1Q3. The circuit breaker in D1Q1 shall be out of the scope of the considered SA system.

D.1.2 Communication system configuration

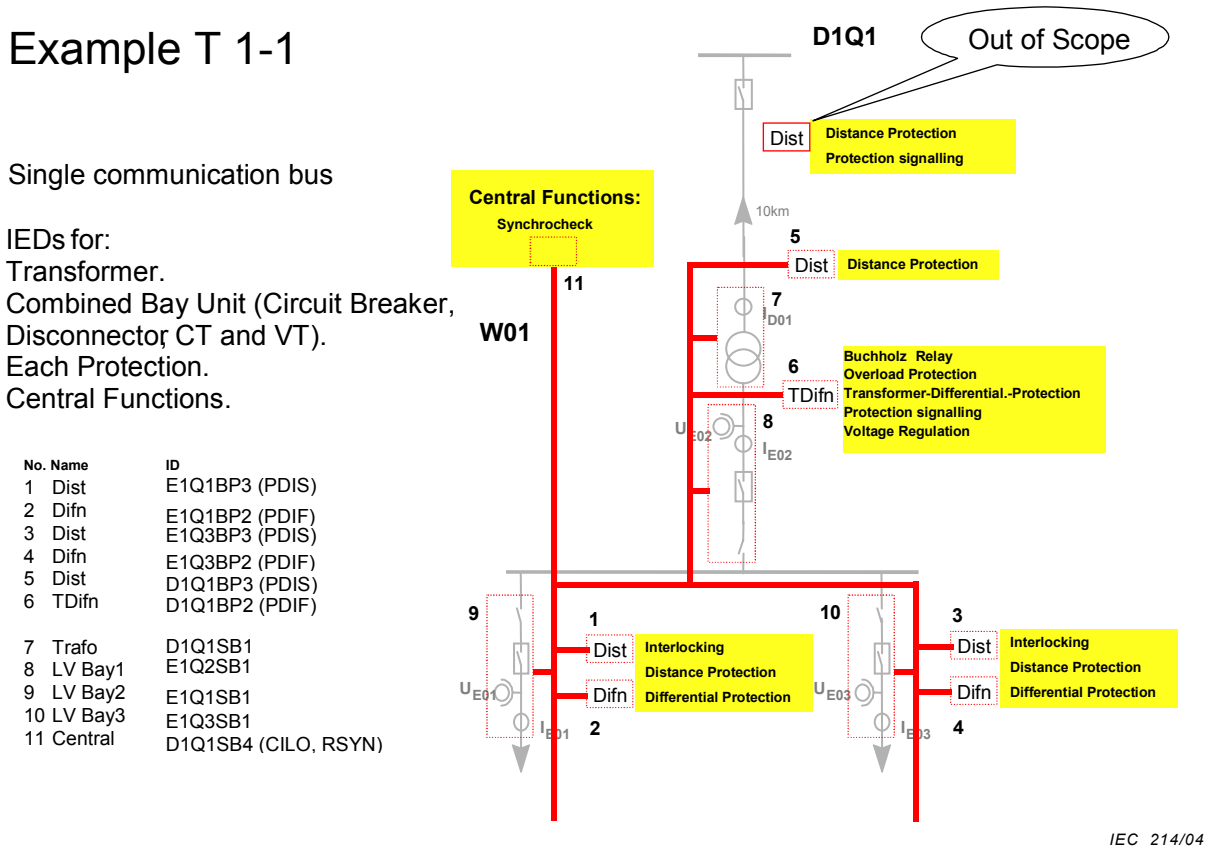


Figure D.2 – T1-1 Communication configuration

Figure D.2 shows the IEDs of the SA system, their allocation to the switchyard bays, their intended functionality, and their communication connection by one single Subnetwork. What is not shown is the IED hosting the station level HMI, which might be a pure client.

D.1.3 Transformer IED

Figure D.3 illustrates for the transformer control IED the instantiated functionality as logical nodes.

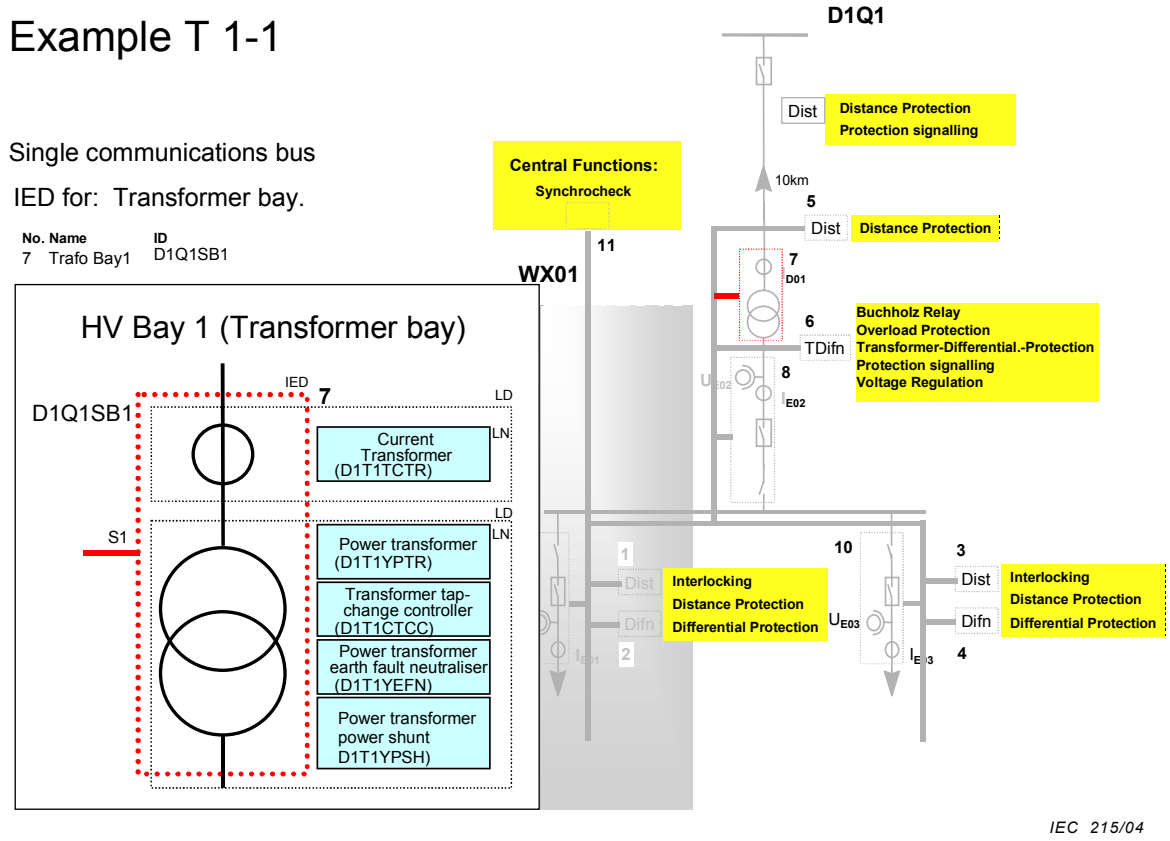


Figure D.3 – T1-1 Transformer bay

D.2 Example SCL file contents

Below is a syntactically correct, but not fully completed SCD file for the example specification given above. For some IEDs the server description is missing, and naturally no data flow from or to these IEDs is specified. On the other hand, some logical nodes which should reside on these IEDs have been allocated to the substation section. Therefore this file is not only incomplete, but also invalid at application level. However, the two IEDs E1Q1SB1 and D1Q1SB4 and some data flow between them with GOOSE and SV is modelled, and the substation topology as such is complete with connection information. The Subnet definition is also complete, at least for the modelled data flow.

```
<?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">
    <Header id="SCL Example T1-1" nameStructure="IEDName"/>
    <Substation name="S12" desc="Baden">
        <VoltageLevel name="D1">
            <PowerTransformer name="T1" type="PTR">
                <LNode lnInst="1" lnClass="PDIF" ldInst="F1" iedName="D1Q1BP2"/>
                <LNode lnInst="1" lnClass="TCTR" ldInst="C1" iedName="D1Q1SB1"/>
                <TransformerWinding name="W1" type="PTW">
                    <Terminal connectivityNode="S12/D1/Q1/L1" substationName="S12" voltageLevelName="D1"
bayName="Q1" cNodeName="L1"/>
                </TransformerWinding>
                <TransformerWinding name="W2" type="PTW">
                    <Terminal connectivityNode="S12/E1/Q2/L3" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L3"/>
                </TransformerWinding>
            </PowerTransformer>
            <Voltage multiplier="k" unit="V">220</Voltage>
            <Bay name="Q1">
                <LNode lnInst="1" lnClass="PDIS" ldInst="F1" iedName="D1Q1BP3"/>
                <ConductingEquipment name="I1" type="CTR">
```

```
                    <Terminal connectivityNode="S12/D1/Q1/L1" substationName="S12" voltageLevelName="D1"
bayName="Q1" cNodeName="L1"/>
                    <SubEquipment name="R" phase="A">
                        <LNode lnClass="TCTR " iedName="D1Q1BP2" ldInst="F1" lnInst="1"/>
                    </SubEquipment>
                    <SubEquipment name="S" phase="B">
                        <LNode lnClass="TCTR " iedName="D1Q1BP2" ldInst="F1" lnInst="2"/>
                    </SubEquipment>
                    <SubEquipment name="T" phase="C">
                        <LNode lnClass="TCTR " iedName="D1Q1BP2" ldInst="F1" lnInst="3"/>
                    </SubEquipment>
                    <SubEquipment name="I0" phase="N">
                        <LNode lnClass="TCTR " iedName="D1Q1BP2" ldInst="F1" lnInst="4"/>
                    </SubEquipment>
                </ConductingEquipment>
                <ConnectivityNode name="L1" pathName="S12/D1/Q1/L1"/>
            </Bay>
        </VoltageLevel>
        <VoltageLevel name="E1">
            <Voltage multiplier="k" unit="V">132</Voltage>
            <Bay name="Q1">
                <LNode lnInst="1" lnClass="MMXU" ldInst="C1" iedName="E1Q1SB1"/>
                <LNode lnInst="1" lnClass="PDIS" ldInst="F1" iedName="E1Q1BP3"/>
                <LNode lnInst="1" lnClass="PDIF" ldInst="F1" iedName="E1Q1BP2"/>
                <ConductingEquipment name="QA1" type="CBR">
                    <LNode lnInst="1" lnClass="CSWI" ldInst="C1" iedName="E1Q1SB1"/>
                    <Terminal connectivityNode="S12/E1/Q1/L1" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L1"/>
                    <Terminal connectivityNode="S12/E1/Q1/L2" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L2"/>
                </ConductingEquipment>
                <ConductingEquipment name="QB1" type="DIS">
                    <Terminal connectivityNode="S12/E1/W1/BB1" substationName="S12" voltageLevelName="E1"
bayName="W1" cNodeName="BB1"/>
                    <Terminal connectivityNode="S12/E1/Q1/L1" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L1"/>
                </ConductingEquipment>
                <ConductingEquipment name="U1" type="VTR">
                    <Terminal connectivityNode="S12/E1/Q1/L2" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L2"/>
                    <Terminal connectivityNode="S12/E1/Q1/L3" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L3"/>
                    <SubEquipment name="A" phase="A">
                        <LNode lnClass="TVTR" iedName="E1Q1SB1" ldInst="C1" lnInst="1" desc="VT phase L1"/>
                    </SubEquipment>
                </ConductingEquipment>
                <ConductingEquipment name="I1" type="CTR">
                    <Terminal connectivityNode="S12/E1/Q1/L3" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L3"/>
                    <Terminal connectivityNode="S12/E1/Q1/L4" substationName="S12" voltageLevelName="E1"
bayName="Q1" cNodeName="L4"/>
                </ConductingEquipment>
                <ConnectivityNode name="L1" pathName="S12/E1/Q1/L1"/>
                <ConnectivityNode name="L2" pathName="S12/E1/Q1/L2"/>
                <ConnectivityNode name="L3" pathName="S12/E1/Q1/L3"/>
                <ConnectivityNode name="L4" pathName="S12/E1/Q1/L4"/>
            </Bay>
            <Bay name="Q2" desc="Turgi">
                <ConductingEquipment name="QA1" type="CBR">
                    <LNode lnInst="1" lnClass="CILO" ldInst="C1" iedName="D1Q1SB4"/>
                    <Terminal connectivityNode="S12/E1/Q2/L0" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L0"/>
                    <Terminal connectivityNode="S12/E1/Q2/L1" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L1"/>
                </ConductingEquipment>
                <ConductingEquipment name="QB1" type="DIS">
                    <LNode lnInst="2" lnClass="CSWI" ldInst="C1" iedName="E1Q2SB1"/>
                    <LNode lnInst="2" lnClass="CILO" ldInst="C1" iedName="D1Q1SB4"/>
                    <Terminal connectivityNode="S12/E1/Q4/B1" substationName="S12" voltageLevelName="E1"
bayName="Q4" cNodeName="B1"/>
                    <Terminal connectivityNode="S12/E1/Q2/L0" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L0"/>
```

```
                </ConductingEquipment>
                <ConductingEquipment name="I1" type="CTR">
                    <Terminal connectivityNode="S12/E1/Q2/L1" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L1"/>
                    <Terminal connectivityNode="S12/E1/Q2/L2" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L2"/>
                </ConductingEquipment>
                <ConductingEquipment name="U1" type="VTR">
                    <Terminal connectivityNode="S12/E1/Q2/L2" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L2"/>
                    <Terminal connectivityNode="S12/E1/Q2/L3" substationName="S12" voltageLevelName="E1"
bayName="Q2" cNodeName="L3"/>
                </ConductingEquipment>
                <ConnectivityNode name="L0" pathName="S12/E1/Q2/L0"/>
                <ConnectivityNode name="L1" pathName="S12/E1/Q2/L1"/>
                <ConnectivityNode name="L2" pathName="S12/E1/Q2/L2"/>
                <ConnectivityNode name="L3" pathName="S12/E1/Q2/L3"/>
                <ConnectivityNode name="L4" pathName="S12/E1/Q2/L4"/>
            </Bay>
            <Bay name="Q3" desc="London">
                <LNode lnInst="1" lnClass="MMXU" ldInst="" iedName="E1Q3KA1"/>
                <LNode lnInst="1" lnClass="PDIS" ldInst="" iedName="E1Q3KA3"/>
                <LNode lnInst="1" lnClass="PDIF" ldInst="" iedName="E1Q3KA2"/>
                <ConductingEquipment name="QA1" type="CBR">
                    <LNode lnInst="1" lnClass="CSWI" ldInst="C1" iedName="E1Q3SB1"/>
                    <Terminal connectivityNode="S12/E1/Q3/L1" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L1"/>
                    <Terminal connectivityNode="S12/E1/Q3/L2" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L2"/>
                </ConductingEquipment>
                <ConductingEquipment name="QB1" type="DIS">
                    <Terminal connectivityNode="S12/E1/W1/BB1" substationName="S12" voltageLevelName="E1"
bayName="W1" cNodeName="BB1"/>
                    <Terminal connectivityNode="S12/E1/Q3/L1" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L1"/>
                </ConductingEquipment>
                <ConductingEquipment name="U1" type="VTR">
                    <Terminal connectivityNode="S12/E1/Q3/L2" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L2"/>
                    <Terminal connectivityNode="S12/E1/Q3/L3" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L3"/>
                </ConductingEquipment>
                <ConductingEquipment name="I1" type="CTR">
                    <Terminal connectivityNode="S12/E1/Q3/L3" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L3"/>
                    <Terminal connectivityNode="S12/E1/Q3/L4" substationName="S12" voltageLevelName="E1"
bayName="Q3" cNodeName="L4"/>
                </ConductingEquipment>
                <ConnectivityNode name="L1" pathName="S12/E1/Q3/L1"/>
                <ConnectivityNode name="L2" pathName="S12/E1/Q3/L2"/>
                <ConnectivityNode name="L3" pathName="S12/E1/Q3/L3"/>
                <ConnectivityNode name="L4" pathName="S12/E1/Q3/L4"/>
            </Bay>
            <Bay name="Q4">
                <ConnectivityNode name="B1" pathName="S12/E1/Q4/B1"/>
            </Bay>
            <Bay name="W1">
                <ConnectivityNode name="BB1" pathName="S12/E1/W1/BB1"/>
            </Bay>
        </VoltageLevel>
    </Substation>
    <Communication>
        <SubNetwork name="W01" type="8-MMS">
            <Text>Station bus</Text>
            <BitRate unit="b/s">10</BitRate>
            <ConnectedAP iedName="D1Q1SB4" apName="S1">
                <Address>
                    <P type="IP" xsi:type="tP_IP">10.0.0.11</P>
                    <P type="IP-SUBNET" xsi:type="tP_IP-SUBNET">255.255.255.0</P>
                    <P type="IP-GATEWAY" xsi:type="tP_IP-GATEWAY">10.0.0.101</P>
                    <P type="OSI-TSEL" xsi:type="tP_OSI-TSEL">00000001</P>
                    <P type="OSI-PSEL" xsi:type="tP_OSI-PSEL">01</P>
```

```
                <P type="OSI-SSEL" xsi:type="tP_OSI-SSEL">01</P>
            </Address>
            <GSE ldInst="C1" cbName="SyckResult">
                <Address>
                    <P type="MAC-Address">01-0C-CD-01-00-02</P>
                    <P type="APPID">3001</P>
                    <P type="VLAN-PRIORITY">4</P>
                </Address>
            </GSE>
            <PhysConn type="Plug">
                <P type="Type">FOC</P>
                <P type="Plug">ST</P>
            </PhysConn>
        </ConnectedAP>
        <ConnectedAP iedName="E1Q1SB1" apName="S1">
            <Address>
                <P type="IP">10.0.0.1</P>
                <P type="IP-SUBNET">255.255.255.0</P>
                <P type="IP-GATEWAY">10.0.0.101</P>
                <P type="OSI-TSEL">00000001</P>
                <P type="OSI-PSEL">01</P>
                <P type="OSI-SSEL">01</P>
            </Address>
            <GSE ldInst="C1" cbName="ItlPositions">
                <Address>
                    <P type="MAC-Address">01-0C-CD-01-00-01</P>
                    <P type="APPID">3000</P>
                    <P type="VLAN-PRIORITY">4</P>
                </Address>
            </GSE>
            <SMV ldInst="C1" cbName="Volt">
                <Address>
                    <P type="MAC-Address">01-0C-CD-04-00-01</P>
                    <P type="APPID">4000</P>
                    <P type="VLAN-ID">123</P>
                    <P type="VLAN-PRIORITY">4</P>
                </Address>
            </SMV>
        </ConnectedAP>
        <ConnectedAP iedName="E1Q1BP2" apName="S1">
            <Address>
                <P type="IP">10.0.0.2</P>
                <P type="IP-SUBNET">255.255.255.0</P>
                <P type="IP-GATEWAY">10.0.0.101</P>
                <P type="OSI-TSEL">00000001</P>
                <P type="OSI-PSEL">01</P>
                <P type="OSI-SSEL">01</P>
            </Address>
        </ConnectedAP>
        <ConnectedAP iedName="E1Q1BP3" apName="S1">
            <Address>
                <P type="IP">10.0.0.3</P>
                <P type="IP-SUBNET">255.255.255.0</P>
                <P type="IP-GATEWAY">10.0.0.101</P>
                <P type="OSI-TSEL">00000001</P>
                <P type="OSI-PSEL">01</P>
                <P type="OSI-SSEL">01</P>
            </Address>
        </ConnectedAP>
    </SubNetwork>
</Communication>
<IED name="E1Q1SB1">
    <Services>
        <DynAssociation/>
        <GetDirectory/>
        <GetDataObjectDefinition/>
        <GetDataSetValue/>
        <DataSetDirectory/>
        <ReadWrite/>
        <FileHandling/>
        <ConfDataSet max="4" maxAttributes="50"/>
        <ConfReportControl max="12"/>
```

```
            <ReportSettings bufTime="Dyn" cbName="Conf" rptID="Dyn" datSet="Conf" intgPd="Dyn"
optFields="Conf"/>
            <ConfLogControl max="1"/>
            <ConfLNs fixLnInst="true"/>
            <GetCBValues/>
            <GOOSE max="2"/>
            <GSESettings appID="Conf" cbName="Conf" datSet="Conf"/>
        </Services>
        <AccessPoint name="S1">
            <Server>
                <Authentication/>
                <LDevice inst="C1">
                    <LN0 lnType="LN0" lnClass="LLN0" inst="">
                        <DataSet name="Positions">
                            <FCDA ldInst="C1" prefix="" lnInst="1" lnClass="CSWI" doName="Pos" fc="ST"/>
                            <FCDA ldInst="C1" prefix="" lnInst="2" lnClass="CSWI" doName="Pos" fc="ST"/>
                        </DataSet>
                        <DataSet name="Measurands">
                            <FCDA ldInst="C1" prefix="" lnInst="1" lnClass="MMXU" doName="Amps" fc="MX"/>
                            <FCDA ldInst="C1" prefix="" lnInst="1" lnClass="MMXU" doName="Volts" fc="MX"/>
                        </DataSet>
                        <DataSet name="smv">
                            <FCDA ldInst="C1" prefix="" lnClass="TVTR" lnInst="1" doName="Vol" daName="instMag"
fc="MX"/>
                        </DataSet>
                        <ReportControl name="PosReport" rptID="E1Q1Switches" datSet="Positions" confRev="0">
                            <TrgOps dchg="true" qchg="true"/>
                            <OptFields/>
                            <RptEnabled max="5">
                                <ClientLN iedName="A1KA1" ldInst="LD1" lnInst="1" lnClass="IHMI"/>
                            </RptEnabled>
                        </ReportControl>
                        <ReportControl name="MeaReport" rptID="E1Q1Measurands" datSet="Measurands"
intgPd="2000" confRev="0">
                            <TrgOps qchg="true" period="true"/>
                            <OptFields reasonCode="true"/>
                            <RptEnabled max="5">
                                <ClientLN iedName="A1KA1" ldInst="LD1" lnInst="1" lnClass="IHMI"/>
                            </RptEnabled>
                        </ReportControl>
                        <LogControl name="Log" datSet="Positions" logName="C1">
                            <TrgOps dchg="true" qchg="true"/>
                        </LogControl>
                        <GSEControl name="ItlPositions" datSet="Positions" appID="Itl"/>
                        <SampledValueControl name="Volt" datSet="smv" smvID="11" smpRate="4800" nofASDU="5"
multicast="true">
                            <SmvOpts sampleRate="true" refreshTime="true" sampleSynchronized="true"/>
                        </SampledValueControl>
                    </LN0>
                    <LN lnType="LPHDa" lnClass="LPHD" inst="1">
                        <DOI name="Proxy">
                            <DAI name="stVal">
                                <Val>false</Val>
                            </DAI>
                        </DOI>
                    </LN>
                    <LN inst="1" lnClass="CSWI" lnType="CSWIa"/>
                    <LN inst="2" lnClass="CSWI" lnType="CSWIa"/>
                    <LN inst="1" lnClass="MMXU" lnType="MMXUa">
                        <DOI name="Volts">
                            <SDI name="sVC">
                                <DAI name="offset">
                                    <Val>10</Val>
                                </DAI>
                                <DAI name="scaleFactor">
                                    <Val>200</Val>
                                </DAI>
                            </SDI>
                        </DOI>
                    </LN>
                    <LN lnType="TVTRa" lnClass="TVTR" inst="1"/>
                </LDevice>
```

```
            </Server>
        </AccessPoint>
    </IED>
    <IED name="E1Q1BP2">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q1BP3">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q2SB1">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q3SB1">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q3KA1">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q3KA2">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="E1Q3KA3">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="D1Q1SB1">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="D1Q1BP2">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="D1Q1BP3">
        <AccessPoint name="S1"/>
    </IED>
    <IED name="D1Q1SB4">
        <Services>
            <DynAssociation/>
            <GetDirectory/>
            <GetDataObjectDefinition/>
            <GetDataSetValue/>
            <DataSetDirectory/>
            <ReadWrite/>
            <FileHandling/>
            <ConfDataSet max="4"/>
            <ConfReportControl max="12"/>
            <ReportSettings bufTime="Dyn" cbName="Conf" rptID="Dyn" datSet="Conf" intgPd="Dyn"
optFields="Conf"/>
            <ConfLogControl max="1"/>
            <GetCBValues/>
            <GOOSE max="2"/>
            <GSESettings appID="Conf" cbName="Conf" datSet="Conf"/>
        </Services>
        <AccessPoint name="S1">
            <Server>
                <Authentication/>
                <LDevice inst="C1">
                    <LN0 lnType="LN0" lnClass="LLN0" inst="">
                        <DataSet name="SyckResult">
                            <FCDA ldInst="C1" prefix="" lnInst="1" lnClass="RSYN" doName="Rel" fc="ST"/>
                        </DataSet>
                        <GSEControl name="SyckResult" datSet="SyckResult" appID="SynChk"/>
                    </LN0>
                    <LN lnType="LPHDa" lnClass="LPHD" inst="1">
                        <DOI name="Proxy">
                            <DAI name="stVal">
                                <Val>false</Val>
                            </DAI>
                        </DOI>
                    </LN>
                    <LN inst="1" lnClass="RSYN" lnType="RSYNa"/>
                </LDevice>
            </Server>
        </AccessPoint>
```

```
    </IED>
    <DataTypeTemplates>
        <LNodeType id="LN0" lnClass="LLN0">
            <DO name="Mod" type="myMod"/>
            <DO name="Health" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="NamPlt" type="myLPL"/>
        </LNodeType>
        <LNodeType id="LPHDa" lnClass="LPHD">
            <DO name="Mod" type="myMod"/>
            <DO name="Health" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="NamPlt" type="myLPL"/>
            <DO name="PhyNam" type="myDPL"/>
            <DO name="PhyHealth" type="myINS"/>
            <DO name="Proxy" type="mySPS"/>
        </LNodeType>
        <LNodeType id="CSWIa" lnClass="CSWI">
            <DO name="Mod" type="myMod"/>
            <DO name="Health" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="Pos" type="myPos"/>
            <DO name="GrpAl" type="mySPS"/>
        </LNodeType>
        <LNodeType id="MMXUa" lnClass="MMXU">
            <DO name="Mod" type="myMod"/>
            <DO name="Beh" type="myHealth"/>
            <DO name="Health" type="myBeh"/>
            <DO name="Amps" type="myMV"/>
            <DO name="Volts" type="myMV"/>
        </LNodeType>
        <LNodeType id="CILOa" lnClass="CILO">
            <DO name="Mod" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="Health" type="myINS"/>
            <DO name="EnaOpen" type="mySPS"/>
            <DO name="EnaClose" type="mySPS"/>
        </LNodeType>
        <LNodeType id="TVTRa" lnClass="TVTR">
            <DO name="Mod" type="myMod"/>
            <DO name="Health" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="Vol" type="mySAV"/>
        </LNodeType>
        <LNodeType id="RSYNa" lnClass="RSYN">
            <DO name="Mod" type="myMod"/>
            <DO name="Health" type="myHealth"/>
            <DO name="Beh" type="myBeh"/>
            <DO name="NamPlt" type="myLPL"/>
            <DO name="Rel" type="mySPS"/>
        </LNodeType>
        <DOType id="myMod" cdc="INC">
            <DA name="ctlVal" fc="CO" bType="Enum" type="Mod"/>
            <DA name="stVal" fc="ST" dchg="true" bType="Enum" type="Mod"/>
            <DA name="q" fc="ST" bType="Quality" dchg="true"/>
            <DA name="t" fc="ST" bType="Timestamp" dchg="true"/>
        </DOType>
        <DOType id="myHealth" cdc="INS">
            <DA name="stVal" fc="ST" bType="Enum" dchg="true" type="Health"/>
        </DOType>
        <DOType id="myBeh" cdc="INS">
            <DA name="stVal" fc="ST" bType="Enum" dchg="true" type="Beh"/>
        </DOType>
        <DOType id="myINS" cdc="INS">
            <DA name="stVal" fc="ST" bType="INT32" dchg="true"/>
        </DOType>
        <DOType id="myLPL" cdc="LPL">
            <DA name="IdNs" fc="EX" bType="VisString255">
                <Val>IEC61850-7-4:2003</Val>
            </DA>
            <DA name="configRev" fc="DC" bType="VisString255">
                <Val>Rev 3.45</Val>
```

```
    </DA>
</DOType>
<DOType id="myDPL" cdc="DPL">
    <DA name="vendor" fc="DC" bType="VisString255">
        <Val>myVendorName</Val>
    </DA>
    <DA name="hwRev" fc="DC" bType="VisString255">
        <Val>Rev 1.23</Val>
    </DA>
</DOType>
<DOType id="myPos" cdc="DPC">
    <DA name="stVal" fc="ST" bType="Dbpos" dchg="true" type="Dbpos"/>
    <DA name="q" fc="ST" bType="Quality" qchg="true"/>
    <DA name="t" fc="ST" bType="Timestamp"/>
    <DA name="ctlVal" fc="CO" bType="BOOL"/>
</DOType>
<DOType id="mySPS" cdc="SPS">
    <DA name="stVal" fc="ST" bType="INT32" dchg="true"/>
    <DA name="q" fc="ST" bType="Quality" qchg="true"/>
    <DA name="t" fc="ST" bType="Timestamp"/>
</DOType>
<DOType id="myMV" cdc="MV">
    <DA name="mag" fc="MX" bType="Struct" type="myAnalogValue" dchg="true"/>
    <DA name="q" fc="MX" bType="Quality" qchg="true"/>
    <DA name="t" fc="MX" bType="Timestamp"/>
    <DA name="sVC" fc="CF" bType="Struct" type="ScaledValueConfig" dchg="true"/>
</DOType>
<DOType id="myCMV" cdc="CMV">
    <DA name="cVal" fc="MX" bType="Struct" type="myVector" dchg="true"/>
    <DA name="q" fc="MX" bType="Quality" qchg="true"/>
    <DA name="t" fc="MX" bType="Timestamp"/>
</DOType>
<DOType id="mySEQ" cdc="SEQ">
    <SDO name="c1" type="myCMV"/>
    <SDO name="c2" type="myCMV"/>
    <SDO name="c3" type="myCMV"/>
    <DA name="seqT" fc="MX" bType="Enum" type="seqT"/>
</DOType>
<DOType id="mySAV" cdc="SAV">
    <DA name="instMag" fc="MX" bType="Struct" type="myAnalogValue"/>
    <DA name="q" fc="MX" bType="Quality" qchg="true"/>
</DOType>
<DAType id="myAnalogValue">
    <BDA name="f" bType="FLOAT32"/>
</DAType>
<DAType id="ScaledValueConfig">
    <BDA name="scaleFactor" bType="FLOAT32"/>
    <BDA name="offset" bType="FLOAT32"/>
</DAType>
<DAType id="myVector">
    <BDA name="mag" bType="Struct" type="myAnalogValue"/>
    <BDA name="ang" bType="Struct" type="myAnalogValue"/>
</DAType>
<EnumType id="ACDdir">
    <EnumVal ord="0">unknown</EnumVal>
    <EnumVal ord="1">forward</EnumVal>
    <EnumVal ord="2">backward</EnumVal>
    <EnumVal ord="3">both</EnumVal>
</EnumType>
<EnumType id="seqT">
    <EnumVal ord="0">pos-neg-zero</EnumVal>
    <EnumVal ord="1">dir-quad-zero</EnumVal>
</EnumType>
<EnumType id="Dbpos">
    <EnumVal ord="0">intermediate</EnumVal>
    <EnumVal ord="1">off</EnumVal>
    <EnumVal ord="2">on</EnumVal>
    <EnumVal ord="3">bad</EnumVal>
</EnumType>
<EnumType id="Tcmd">
    <EnumVal ord="0">stop</EnumVal>
    <EnumVal ord="1">lower</EnumVal>
```

```
            <EnumVal ord="2">higher</EnumVal>
            <EnumVal ord="3">reserved</EnumVal>
        </EnumType>
        <EnumType id="Beh">
            <EnumVal ord="1">on</EnumVal>
            <EnumVal ord="2">blocked</EnumVal>
            <EnumVal ord="3">test</EnumVal>
            <EnumVal ord="4">test/blocked</EnumVal>
            <EnumVal ord="5">off</EnumVal>
        </EnumType>
        <EnumType id="Mod">
            <EnumVal ord="1">on</EnumVal>
            <EnumVal ord="2">blocked</EnumVal>
            <EnumVal ord="3">test</EnumVal>
            <EnumVal ord="4">test/blocked</EnumVal>
            <EnumVal ord="5">off</EnumVal>
        </EnumType>
        <EnumType id="Health">
            <EnumVal ord="1">Ok</EnumVal>
            <EnumVal ord="2">Warning</EnumVal>
            <EnumVal ord="3">Alarm</EnumVal>
        </EnumType>
    </DataTypeTemplates>
</SCL>
```

Annex E
(informative)

XML schema definition of SCL variants

The following schema part, which uses elements from the normative SCL schema definition, is however not itself normative and formally defines the restrictions for the different SCL variants introduced in Clause 7:

CID: Configured IED Description

ICD: IED Capability Description

SCD: Substation Configuration Description

SSD: Substation Specification Description; here a 'pure' version without IEDs, and a version with some already known IEDs is introduced.

Observe that additionally to the restrictions formulated here, some naming restrictions as described in Clause 7 apply, which cannot be expressed in XML schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" finalDefault="extension" version="1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            COPYRIGHT IEC, 2003. Version 1.0. Release 2003/08/20.
            This schema is for infomational purposes only, and is not normative!
            Notes:
            – Identity constraints are in comments, in order to avoid any clashes with the existing ones.
            – The elements are defined as abstract to prevent their usage in practice.
        </xs:documentation>
    </xs:annotation>
    <!-- ========================================
            Including the general case:
        ======================================== -->
    <xs:include schemaLocation="SCL.xsd"/>
    <!-- ========================================
            IED Capability Description (ICD) variant
        ======================================== -->
    <xs:element name="SCL_ICD" abstract="true">
        <xs:annotation>
            <xs:documentation xml:lang="en">SCL for an IED Capability Description (ICD)</xs:documentation>
        </xs:annotation>
        <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 name="Substation" type="tSubstationTemplate" minOccurs="0">
                            <!--<xs:unique name="uniqueVoltageLevelInSubstation">
                                <xs:selector xpath="./scl:VoltageLevel"/>
                                <xs:field xpath="@name"/>
                            </xs:unique>
                            <xs:unique name="uniquePowerTranformerInSubstation">
                                <xs:selector xpath="./scl:PowerTransformer"/>
                                <xs:field xpath="@name"/>
                            </xs:unique>
                            <xs:unique name="uniqueFunctionInSubstation">
                                <xs:selector xpath="./scl:Function"/>
```

```
                        <xs:field xpath="@name"/>
                    </xs:unique>
                    <xs:key name="ConnectivityNodeKey">
                        <xs:selector xpath=".//scl:ConnectivityNode"/>
                        <xs:field xpath="@pathName"/>
                    </xs:key>
                    <xs:keyref name="ref2ConnectivityNode" refer="ConnectivityNodeKey">
                        <xs:selector xpath=".//scl:Terminal"/>
                        <xs:field xpath="@connectivityNode"/>
                    </xs:keyref>
                    <xs:unique name="uniqueLNode">
                        <xs:selector xpath=".//scl:LNode"/>
                        <xs:field xpath="@lnInst"/>
                        <xs:field xpath="@lnClass"/>
                        <xs:field xpath="@iedName"/>
                        <xs:field xpath="@ldInst"/>
                        <xs:field xpath="@prefix"/>
                    </xs:unique>-->
                </xs:element>
                <xs:element ref="Communication" minOccurs="0"/>
                <xs:element name="IED" type="tIEDTemplate">
                    <!--<xs:unique name="uniqueAccessPointInIED">
                        <xs:selector xpath="./scl:AccessPoint"/>
                        <xs:field xpath="@name"/>
                    </xs:unique>
                    <xs:unique name="uniqueLDeviceInIED">
                        <xs:selector xpath=".//scl:LDevice"/>
                        <xs:field xpath="@inst"/>
                    </xs:unique>
                    <xs:unique name="uniqueGSEControlInIED">
                        <xs:selector xpath=".//scl:GSEControl"/>
                        <xs:field xpath="@name"/>
                    </xs:unique>
                    <xs:unique name="uniqueSMVControlInIED">
                        <xs:selector xpath=".//scl:SampledValueControl"/>
                        <xs:field xpath="@name"/>
                    </xs:unique>
                    <xs:key name="LDeviceInIEDKey">
                        <xs:selector xpath="./scl:AccessPoint/scl:Server/scl:LDevice"/>
                        <xs:field xpath="@inst"/>
                    </xs:key>
                    <xs:keyref name="ref2LDeviceInIED" refer="LDeviceInIEDKey">
                        <xs:selector xpath="./scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:LogControl"/>
                        <xs:field xpath="@logName"/>
                    </xs:keyref>-->
                </xs:element>
                <xs:element ref="DataTypeTemplates"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!--<xs:key name="LNodeTypeKey">
    <xs:selector xpath="./scl:DataTypeTemplates/scl:LNodeType"/>
    <xs:field xpath="@id"/>
    <xs:field xpath="@lnClass"/>
</xs:key>
<xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
    <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
    <xs:field xpath="@lnType"/>
    <xs:field xpath="@lnClass"/>
</xs:keyref>
<xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
    <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
    <xs:field xpath="@lnType"/>
    <xs:field xpath="@lnClass"/>
</xs:keyref>
<xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
    <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
    <xs:field xpath="@lnType"/>
    <xs:field xpath="@lnClass"/>
</xs:keyref>-->
</xs:element>
```

```
    <!-- ========================================
            "Pure" System Specification Document (SSD) variant
         ======================================== -->
    <xs:element name="SCL_pureSSD" abstract="true">
        <xs:annotation>
            <xs:documentation xml:lang="en">SCL for a "Pure" System Specification Document
(SSD)</xs:documentation>
        </xs:annotation>
        <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" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!--<xs:unique name="uniqueSubstation">
            <xs:selector xpath="./scl:Substation"/>
            <xs:field xpath="@name"/>
        </xs:unique>-->
    </xs:element>
    <!-- ========================================
            System Specification Document (SSD) variant
         ======================================== -->
    <xs:element name="SCL_SSD" abstract="true">
        <xs:annotation>
            <xs:documentation xml:lang="en">SCL for a System Specification Document (SSD)</xs:documentation>
        </xs:annotation>
        <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" maxOccurs="unbounded"/>
                        <xs:element ref="Communication" minOccurs="0"/>
                        <xs:element ref="IED" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element ref="DataTypeTemplates" minOccurs="0"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!--<xs:unique name="uniqueSubstation">
            <xs:selector xpath="./scl:Substation"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="IEDKey">
            <xs:selector xpath="./scl:IED"/>
            <xs:field xpath="@name"/>
        </xs:key>
        <xs:key name="LNodeTypeKey">
            <xs:selector xpath="./scl:DataTypeTemplates/scl:LNodeType"/>
            <xs:field xpath="@id"/>
            <xs:field xpath="@lnClass"/>
        </xs:key>
        <xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
```

```
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>-->
    </xs:element>
    <!-- ========================================
            Substation Configuration Description (SCD) variant
        ======================================== -->
    <xs:element name="SCL_SCD" abstract="true">
        <xs:annotation>
            <xs:documentation xml:lang="en">SCL for a Substation Configuration Description
(SCD)</xs:documentation>
        </xs:annotation>
        <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" maxOccurs="unbounded"/>
                        <xs:element ref="Communication"/>
                        <xs:element ref="IED" maxOccurs="unbounded"/>
                        <xs:element ref="DataTypeTemplates"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!--<xs:unique name="uniqueSubstation">
            <xs:selector xpath="./scl:Substation"/>
            <xs:field xpath="@name"/>
        </xs:unique>
        <xs:key name="IEDKey">
            <xs:selector xpath="./scl:IED"/>
            <xs:field xpath="@name"/>
        </xs:key>
        <xs:key name="LNodeTypeKey">
            <xs:selector xpath="./scl:DataTypeTemplates/scl:LNodeType"/>
            <xs:field xpath="@id"/>
            <xs:field xpath="@lnClass"/>
        </xs:key>
        <xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>-->
    </xs:element>
    <!-- ========================================
            Configured IED Description (CID) variant
        ======================================== -->
    <xs:element name="SCL_CID" abstract="true">
```

```
        <xs:annotation>
            <xs:documentation xml:lang="en">SCL for a Configured IED Description (CID)</xs:documentation>
        </xs:annotation>
        <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"/>
                        <xs:element ref="IED" maxOccurs="unbounded"/>
                        <xs:element ref="DataTypeTemplates"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!--<xs:key name="LNodeTypeKey">
            <xs:selector xpath="./scl:DataTypeTemplates/scl:LNodeType"/>
            <xs:field xpath="@id"/>
            <xs:field xpath="@lnClass"/>
        </xs:key>
        <xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>
        <xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
            <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
            <xs:field xpath="@lnType"/>
            <xs:field xpath="@lnClass"/>
        </xs:keyref>-->
    </xs:element>
    <!-- ========================================
         Miscellaneous type restrictions
         ======================================== -->
    <xs:complexType name="tSubstationTemplate">
        <xs:complexContent>
            <xs:restriction base="tSubstation">
                <xs:sequence>
                    <xs:sequence>
                        <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="Text" type="tText" minOccurs="0"/>
                        <xs:element name="Private" type="tPrivate" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:sequence>
                        <xs:element name="LNode" type="tLNode" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                    <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:sequence>
                    <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: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:sequence>
            <xs:attribute name="name" type="tName" use="required" fixed="TEMPLATE"/>
        </xs:restriction>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tIEDTemplate">
    <xs:complexContent>
        <xs:restriction base="tIED">
            <xs:sequence>
                <xs:sequence>
                    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="Text" type="tText" minOccurs="0"/>
                    <xs:element name="Private" type="tPrivate" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:sequence>
                    <xs:element name="Services" type="tServices" minOccurs="0"/>
                    <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
                        <!--<xs:unique name="uniqueLNInAccessPoint">
                            <xs:annotation>
                                <xs:documentation xml:lang="en">Only for those LN that are direct children of this
AccessPoint.</xs:documentation>
                            </xs:annotation>
                            <xs:selector xpath=".//scl:LN"/>
                            <xs:field xpath="@inst"/>
                            <xs:field xpath="@lnClass"/>
                            <xs:field xpath="@prefix"/>
                        </xs:unique>-->
                    </xs:element>
                </xs:sequence>
            </xs:sequence>
            <xs:attribute name="name" type="tName" use="required" fixed="TEMPLATE"/>
        </xs:restriction>
    </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Bibliography

UML™ Resource Page, OMG, available at: <http://www.omg.org/uml>

Standards Survey

The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!

Customer Service Centre (CSC)

International Electrotechnical Commission
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

A Prioritaire

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

RÉPONSE PAYÉE

SUISSE

Customer Service Centre (CSC)
International Electrotechnical Commission
3, rue de Varembé
1211 GENEVA 20
Switzerland

Q1 Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

.....

Q2 Please tell us in what capacity(ies) you bought the standard *(tick all that apply).* I am the/a:

purchasing agent ❑
librarian ❑
researcher ❑
design engineer ❑
safety engineer ❑
testing engineer ❑
marketing specialist ❑
other.....

Q3 I work for/in/as a: *(tick all that apply)*

manufacturing ❑
consultant ❑
government ❑
test/certification facility ❑
public utility ❑
education ❑
military ❑
other.....

Q4 This standard will be used for: *(tick all that apply)*

general reference ❑
product research ❑
product design/development ❑
specifications ❑
tenders ❑
quality assessment ❑
certification ❑
technical documentation ❑
thesis ❑
manufacturing ❑
other.....

Q5 This standard meets my needs: *(tick one)*

not at all ❑
nearly ❑
fairly well ❑
exactly ❑

Q6 If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

standard is out of date ❑
standard is incomplete ❑
standard is too academic ❑
standard is too superficial ❑
title is misleading ❑
I made the wrong choice ❑
other .....

Q7 Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

timeliness.....
quality of writing.....
technical contents.....
logic of arrangement of contents .....
tables, charts, graphs, figures.....
other .....

Q8 I read/use the: *(tick one)*

French text only ❑
English text only ❑
both English and French texts ❑

Q9 Please share any comment on any aspect of the IEC that you would like us to know:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

ISBN 2-8318-7432-7

ICS 33.200

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND