



**57/919/CDV**

**COMMITTEE DRAFT FOR VOTE (CDV)  
PROJET DE COMITÉ POUR VOTE (CDV)**

Project number Numéro de projet		IEC 61850-6 Ed.2.0	
IEC/TC or SC: <b>TC 57</b> CEI/CE ou SC:		Secretariat / Secrétariat <b>Germany</b>	
<input checked="" type="checkbox"/> Submitted for parallel voting in CENELEC <input type="checkbox"/> Soumis au vote parallèle au CENELEC	Date of circulation Date de diffusion <b>2008-02-08</b>	Closing date for voting (Voting mandatory for P-members) Date de clôture du vote (Vote obligatoire pour les membres (P)) <b>2008-07-11</b>	
Also of interest to the following committees Intéresse également les comités suivants		Supersedes document Remplace le document <b>57/893/CD - 57/915/CC</b>	
Functions concerned Fonctions concernées			
<input type="checkbox"/> Safety Sécurité	<input type="checkbox"/> EMC CEM	<input type="checkbox"/> Environment Environnement	<input type="checkbox"/> Quality assurance Assurance qualité

CE DOCUMENT EST TOUJOURS À L'ÉTUDE ET SUSCEPTIBLE DE MODIFICATION. IL NE PEUT SERVIR DE RÉFÉRENCE.

LES RÉCIPIENDAIRES DU PRÉSENT DOCUMENT SONT INVITÉS À PRÉSENTER, AVEC LEURS OBSERVATIONS, LA NOTIFICATION DES DROITS DE PROPRIÉTÉ DONT ILS AURAIENT ÉVENTUELLEMENT CONNAISSANCE ET À FOURNIR UNE DOCUMENTATION EXPLICATIVE.

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Title :

**IEC 61850-6 Ed.2: Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs**

Introductory note

The French National Committee will not provide a French translation for this project.

**ATTENTION  
VOTE PARALLÈLE  
CEI – CENELEC**

L'attention des Comités nationaux de la CEI, membres du CENELEC, est attirée sur le fait que ce projet de comité pour vote (CDV) de Norme internationale est soumis au vote parallèle.  
Un bulletin de vote séparé pour le vote CENELEC leur sera envoyé par le Secrétariat Central du CENELEC.

**ATTENTION  
IEC – CENELEC  
PARALLEL VOTING**

The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) for an International Standard is submitted for parallel voting.  
A separate form for CENELEC voting will be sent to them by the CENELEC Central Secretariat.

**Copyright © 2008 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.**

## CONTENTS

FOREWORD.....	7
INTRODUCTION.....	9
1 Scope.....	10
2 Normative references.....	10
3 Terms and definitions .....	11
4 Abbreviations.....	12
5 Intended engineering process with SCL.....	13
5.1 General .....	13
5.2 Scope of SCL .....	13
5.3 Use of SCL in the Engineering process .....	14
5.4 IED modifications .....	17
5.5 Data exchange between projects .....	18
6 The SCL object model .....	20
6.1 General .....	20
6.2 The substation model .....	23
6.3 The product (IED) model.....	24
6.4 The communication system model .....	25
6.5 Modelling of redundancy .....	26
7 SCL description file types.....	27
8 The SCL language .....	29
8.1 Specification method .....	29
8.2 Language versions and compatibility.....	31
8.2.1 MustUnderstand rules.....	32
8.2.2 SCL name space and versions.....	32
8.2.3 Incompatibilities to earlier versions.....	33
8.3 SCL language extensions .....	34
8.3.1 General .....	34
8.3.2 Data model extensions.....	34
8.3.3 Additional semantics to existing syntax elements.....	34
8.3.4 Data type constraints .....	34
8.3.5 XML name spaces .....	34
8.3.6 Private data .....	35
8.3.7 Another XML syntax.....	36
8.3.8 Summary: Standard conformance for extension handling.....	36
8.3.9 Extension example.....	36
8.4 General structure.....	36
8.5 Object and signal designation .....	37
8.5.1 General .....	37
8.5.2 Object designations in an object hierarchy.....	38
8.5.3 Signal identifications to be used in the communication system.....	38
8.5.4 Signal identifications usable by applications .....	40
8.5.5 Naming example .....	40
9 The SCL syntax elements .....	41
9.1 Header .....	41
9.2 Substation description .....	44
9.2.1 General .....	44

9.2.2	Voltage level.....	47
9.2.3	Bay level.....	48
9.2.4	Power equipment.....	49
9.2.5	SubEquipment level.....	52
9.2.6	Substation function logical nodes.....	53
9.2.7	Non power equipment.....	54
9.2.8	Substation section example.....	55
9.3	IED description.....	58
9.3.1	General.....	58
9.3.2	The IED, Services and Access Point.....	62
9.3.3	The IED server.....	68
9.3.4	The logical device.....	69
9.3.5	LN0 and other Logical Nodes.....	70
9.3.6	DATA (DOI) definition.....	72
9.3.7	Data set definition.....	74
9.3.8	Report control block.....	76
9.3.9	Log control block.....	80
9.3.10	GSE control block.....	81
9.3.11	Sampled value control block.....	82
9.3.12	Setting control block.....	83
9.3.13	Binding to external signals.....	84
9.3.14	Associations.....	85
9.4	Communication system description.....	86
9.4.1	General.....	86
9.4.2	Subnetwork definition.....	87
9.4.3	Address definition.....	89
9.4.4	GSE address definition.....	90
9.4.5	SMV address definition.....	90
9.4.6	Physical connection parameters.....	91
9.4.7	Communication section example.....	92
9.5	Data type templates.....	92
9.5.1	General.....	92
9.5.2	LNNodeType definitions.....	95
9.5.3	DO type definition.....	96
9.5.4	Data attribute (DA) definition.....	97
9.5.5	Data attribute structure type.....	101
9.5.6	Enumeration types.....	102
9.5.7	Data type template examples.....	103
10	Tool and project engineering rights.....	103
10.1	IED configurator.....	103
10.2	System configurator.....	104
10.3	Right transfer between projects.....	104
Annex A (normative)	SCL syntax: XML schema definition.....	106
A.1	Base types.....	106
A.2	Substation syntax.....	119
A.3	Data type templates.....	126
A.4	IED capabilities and structure.....	129
A.5	Communication subnetworks.....	139

- A.6 Main SCL..... 143
- Annex B (normative) SCL enumerations according to IEC 61850-7-3 and IEC 61850-7-4 .... 145
- Annex C (normative) Syntax extension examples..... 151
  - C.1 Extension syntax for drawing layout coordinates..... 151
  - C.2 Extension syntax for Data model description ..... 153
- Annex D (informative) Example ..... 164
  - D.1 Example specification ..... 164
  - D.2 Example SCL file contents ..... 166
- Annex E (informative) SCL syntax: General XML schema definition ..... 176
- Annex F (informative) XML schema definition of SCL variants ..... 199
- Annex G (normative) SCL Implementation Conformance Statement (SICS)..... 205

Figure 1 – Reference model for information flow in the configuration process.....	15
Figure 2 - IED type description to System configurator.....	16
Figure 3 - IED Instance description to System configurator .....	16
Figure 4 - Modification process.....	17
Figure 5 - Engineering right handling in projects .....	19
Figure 6 – SCL object model .....	21
Figure 7 – SA system configuration example .....	23
Figure 8 – ICD files describing implementable IED types of a general IED class.....	29
Figure 9 – UML diagram overview of SCL schema .....	31
Figure 10 – Elements of the signal identification as defined in IEC 61850-7-2 .....	38
Figure 11 – Elements of the signal name using product naming .....	39
Figure 12 – Possible elements of the signal name using functional naming .....	40
Figure 13 – Names within different structures of the object model.....	41
Figure 14 – UML diagram of Header section .....	42
Figure 15 – UML diagram of Substation section .....	45
Figure 16 – UML diagram for equipment type inheritance and relations.....	49
Figure 17 - Substation section example .....	56
Figure 18 – IED structure and access points.....	59
Figure 19 – UML description of IED related schema part – base .....	60
Figure 20 – UML description of IED related schema part for Control blocks.....	61
Figure 21 – UML description of IED related schema part – LN definition.....	62
Figure 22 – UML diagram overview of the Communication section .....	87
Figure 23 – UML overview of DataTypeTemplate section .....	93
Figure 24 Schema overview .....	154
Table 1 – The files composing the XML schema definition for SCL.....	30
Table 2 – Attributes of the Private element .....	35
Table 3 – Attributes of the Header element.....	42
Table 4 – Attributes of the History item (Hitem) element .....	43
Table 5 – Primary apparatus device type codes .....	51
Table 6 – Attributes of the Terminal element.....	52
Table 7 – Attributes of the SubEquipment element.....	53
Table 8 – Attributes of the LNode element .....	53
Table 10 – Attributes of the IED element .....	63
Table 11 – List of service capabilities and setting elements and attributes .....	64
Table 12 – Attributes of the Access point element.....	67
Table 13 – Attributes of the IED server element.....	69
Table 14 – Attributes of the Authentication element .....	69
Table 15 – Attributes of the LDevice element.....	70
Table 16 – Attributes of the LN0 element.....	71
Table 17 – Attributes of the LN element.....	72
Table 18 – Attributes of the DOI element .....	73
Table 19 – Attributes of the DAI element .....	73
Table 20 – Attributes of the SDI element .....	74

Table 21 – Attributes of the DataSet element.....	75
Table 22 – Attributes of the FCDA element.....	75
Table 23 – Attributes of the FCCB element.....	75
Table 24 – Attributes of the report control block element.....	77
Table 25 – Attributes of the RptEnabled element .....	78
Table 26 – Attributes of the ClientLN element.....	79
Table 27 – Attributes of the log control block element .....	80
Table 28 – Attributes of the GSE control block element.....	81
Table 29 – Attributes of the sampled value control block element.....	82
Table 30 – Attributes of the Smv Options element .....	83
Table 31 - Deprecated Smv options:.....	83
Table 32 – Attributes of the setting control block element .....	84
Table 33 – Attributes of the Input/ExtRef element .....	84
Table 34 – Attributes of the Association element .....	85
Table 35 – Attributes of the Subnetwork element .....	88
Table 36 – Attributes of the ConnectedAP element .....	88
Table 37 – Attributes of the GSE element.....	90
Table 38 – Attributes of the SMV element.....	91
Table 39 – PhysConn P-Type definitions .....	91
Table 40 – Template definition elements .....	95
Table 41 – Attributes of the LNodeType element.....	95
Table 42 – Attributes of the DO element.....	96
Table 43 – Attributes of the DOType element.....	96
Table 44 – Attributes of the SDO element.....	96
Table 45 – Data type mapping.....	97
Table 46 – Attribute value kind (Valkind) meaning .....	98
Table 47 – Attributes of the DA element .....	99
Table 48 – Attributes of the BDA element.....	101
Table 49 – Attributes of the EnumType element.....	103

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

---

**COMMUNICATION NETWORKS AND SYSTEMS  
FOR POWER UTILITY AUTOMATION –****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.

**Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.**

This CD of the International Standard IEC 61850-6 has been prepared by the working group 10 of IEC technical committee 57.

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 for Power Utility Automation*:

- 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

## 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.

Compared to the published edition 1, this edition 2 of this standard:

- Has functional extensions based on changes in other parts, especially part 7-2
- Has functional extensions concerning the engineering process, especially for configuration data exchange between system configuration tools.
- Provides clarifications and corrections to the published edition 1 of IEC 61850-6. Issues that require clarification are published in a database at [www.tissue.iec61850.com](http://www.tissue.iec61850.com). Arising incompatibilities are listed in clause 8.2.3

Note: the process description part, which is currently restricted to switch yards and general process functions, will be enhanced by appropriate add ons for wind mills, hydro plants and distributed energy resources (DER).

## COMMUNICATION NETWORKS AND SYSTEMS FOR UTILITY AUTOMATION –

### 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*

IEC 62351-6, *Data and Communications Security, Part 6: Security for IEC 61850*

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.

Additionally the following terms are used in the context of language name spaces. Observe that here only the general meaning is given. More details about the handling in the context of SCL can be found later in this document.

**Extensible** A language is extensible if instances of the language can include terms from other vocabularies. This is fulfilled in SCL if the other vocabularies come with their own XML name space.

**Language** a language is an identifiable set of vocabulary terms that has defined constraints. This is the case with SCL, although some constraints are not defineable in the XML schema.

**Instance** An instance of a language is a realization by usage of a language; e.g. an XML document in SCL describing an IED or a substation is an SCL instance.

**Sender** A sender creates or produces an instance for processing by another application (receiver). SCL senders are typically IED and system configuration tools; e.g. the IED tool sends (produces) ICD files, the system tool sends SCD files.

**Receiver** A receiver consumes an instance that it obtained from a sender. SCL receivers are IED tools and system configuration tools; e.g. the IED tool receives SCD files, the system tool ICD, SSD and (part) SCD files.

**Processor** a component which receives SCL instances and produces new instances, i.e. is sender and receiver. This is typically the system configuration tool.

**Project** a system part with engineering responsibility for all contained IEDs. Mostly a system is a project. However, sometimes the IED engineering responsibility of different parts of a system belong to different parties or people. Each IED responsibility area is then a separate project. An IED can belong only to one project. It is 'owned' by this project.

**Backwards compatible** A language change is backwards compatible, if newer receivers can process all instances of the old language. For SCL this means that tools built for newer language versions can understand instances from older versions. Especially system tools should understand old ICD and SSD files, while IED tools should understand old SCD files to be backward compatible.

**Forward compatible** A language change is forward compatible if older receivers can process all instances of the newer language. For SCL this means that tools built according to older SCL versions can also process instances of newer SCL versions. Especially old system tools should handle new ICD and SSD files, while old IED tools should handle new SCD files to be forward compatible.

**Language version** the version of the XML schema defining the language. A language instance is produced according to a language (schema) version, which is called its assigned version, although it may also be valid against other language versions.

## 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, i.e. 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)
ID	Identifier
IED	Intelligent Electronic Device
IdInst	Instance identification of a Logical Device as part of its name
InInst	Instance number of a Logical Node as part of its name
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
SDO	Sub DATA within a DOType, referencing another DOType

UML	Unified Modelling Language according to <a href="http://www.omg.org/uml">http://www.omg.org/uml</a>
URI	Universal Resource Identifier
UsvID	ID for USV (Unicast Sampled Value)
XML	Extensible Markup Language

## 5 Intended engineering process with SCL

### 5.1 General

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, or for a part of an already configured process or automation system.
- 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 either for a typical usage of the IED, or for a specific usage within a project.

### 5.2 Scope of SCL

The scope of SCL as defined in this standard is clearly focussed on 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)

These purposes shall support in a standardised way system design, communication engineering and the description of the readily engineered system communication for the device engineering tools.

For practical purpose also the following is supported:

- 4) Exchange of system interfacing information between two projects handling two systems, which need to exchange data
- 5) Exchange of IED modifications on an IED instance engineered specifically for a project back from the IED tool to the system tool

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 and compatible way.

### 5.3 Use of SCL in the Engineering process

Figure 1 explains the usage of SCL data exchange in the above-mentioned engineering process. The 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.

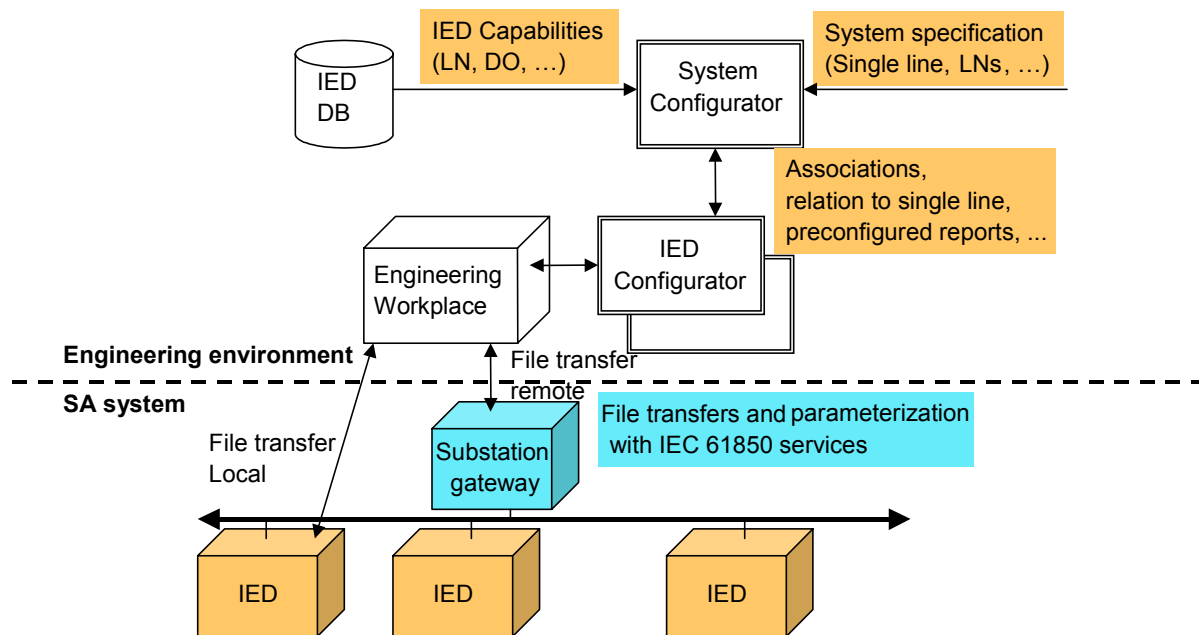
To make the engineering tasks and responsibilities clear, tool roles are introduced for an IED configurator and a system configurator. A 'real' tool can play both roles. In that case the transfer of partly engineered data within the tool is private, but to any other, mostly to an IED tool it has to be seen from the role the tool has played when modifying the project data.

The **IED Configurator** is a manufacturer-specific, may be even IED specific, tool that shall be able to import or export the files defined by this part of IEC 61850. The tool then 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 (ICD) SCL file describing its capabilities, or by an (IID) SCL file describing its project specific configuration and capabilities, or by a tool, which can generate one or both of these file types from or for the IED (not shown in Figure 1).
- It can directly use a system SCL (SCD) file to set its communication configuration, as far as setting is possible in this IED (i.e. as a minimum, its needed communication 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 is 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.



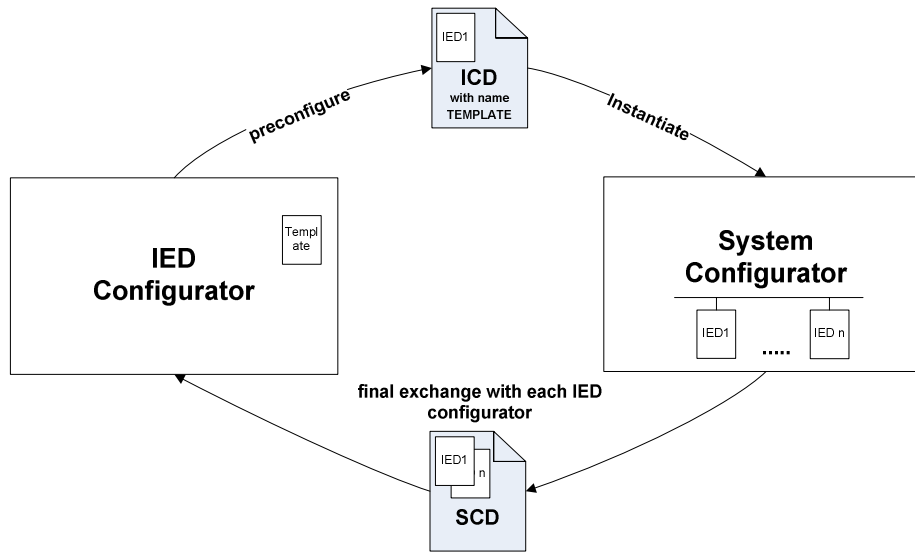
**Figure 1 – Reference model for information flow in the configuration process**

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 communication from an engineering workstation connected locally to the IED. This data 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 SCL format is a possible choice at least of a part of the configuration data.
- 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 conceptual tools respective tool roles 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.

The data exchange between engineering tools during the engineering process can then be as follows:



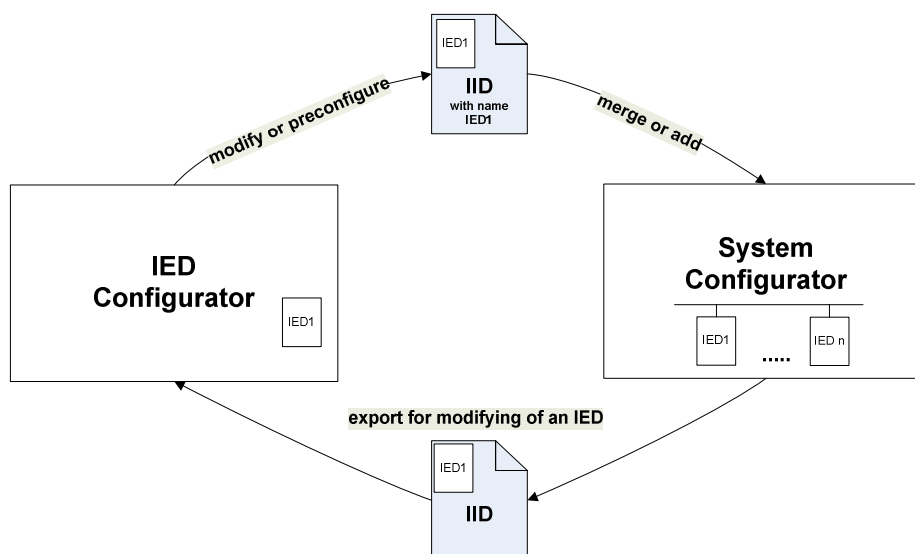
**Figure 2 - IED type description to System configurator**

At start of system engineering the IED capability (ICD) files are used by the system configurator as IED template (type) description to instantiate project specific IEDs as needed. As Figure 2 shows, this ICD file can be generated in advance as typical configuration of an IED by means of the IED configurator tool, as typically done for very flexibly configurable IEDs.

Alternatively the system configurator can also import the description of an IED specifically preconfigured with name and addresses for a concrete function in the process by importing an Instantiated IED Description (IID) file, as indicated in Figure 3.

The SCD file generated by the system configurator tool is then imported by the IED configurator for the final IED instance configuration, as shown in Figure 2 and Figure 3. Any add-ons or adapted values could then be exported by means of the IID file, and thus brought back to the system configurator respective the next revision of the SCD file – see next clause on IED modifications.

To allow detection of a modified file, within the IID file the IED owner shall be set to the SCD header identification of the SCD, which was the input to IED engineering before.



**Figure 3 - IED Instance description to System configurator**

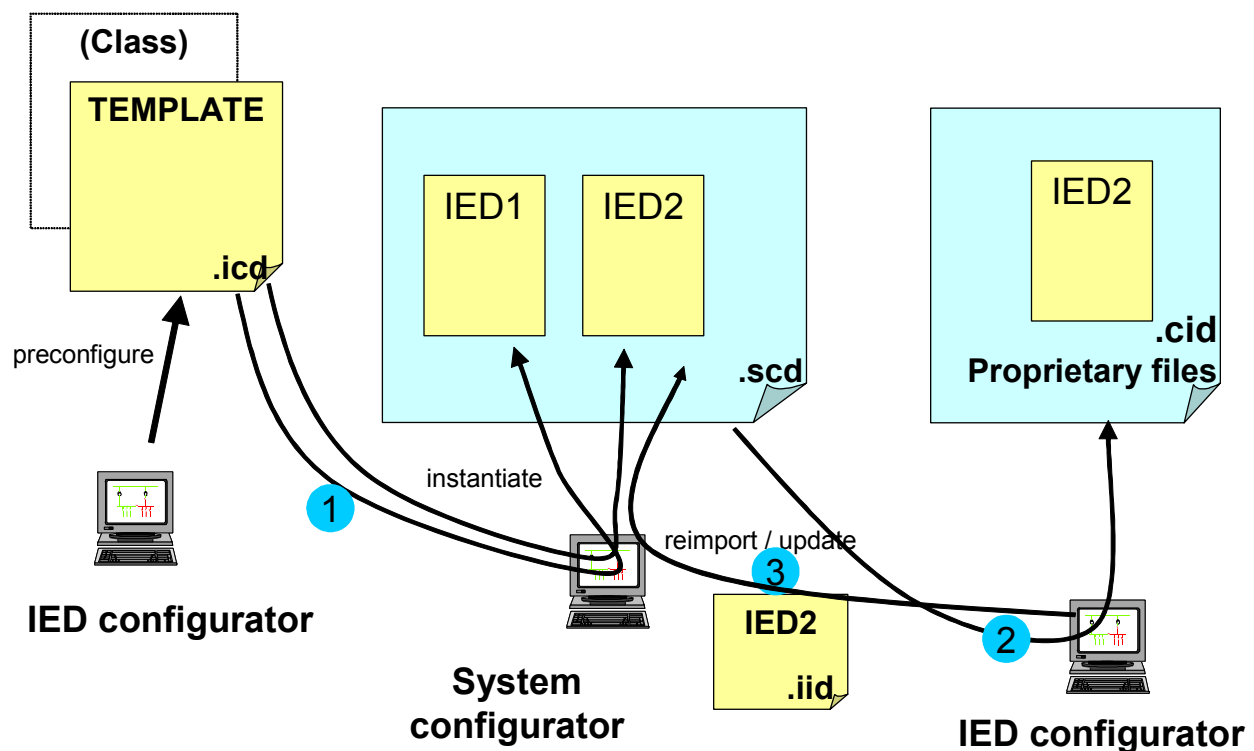
## 5.4 IED modifications

During the engineering process it may happen that the IED related data has to be changed. This can in principle be done by removing the IED from the system, and re-instantiating a modified IED description file in the system. However, in this case also all existing references from or to the IED are lost and have to be re-established. On the other hand, tool responsibilities shall be clear as follows:

The IED configurator is responsible for the IEDs data model, and all its configuration values. It is not allowed to change any data flow and communication related definitions. To assure this, it shall not directly modify a system description (SCD) file.

The System configurator is responsible for the communication addressing and the data flow between the IEDs, within the scope of the IED capabilities. It might set configuration and parameter values as needed from the system point of view. It is not allowed to change the IEDs data model.

Therefore another SCL file is defined, allowing to update the IED data within a system. It is called IID file (Instantiated IED Description), and describes the project specific configuration of an IED. In case that this IED does not exist in the SCD file, it can be imported completely and instantiated as project specific IED, without any references to other IEDs (see also Figure 3). In case that it exists already, the data model part inclusive any values can replace the appropriate parts existing in the system configurator. All data flow related definitions and references to other IEDs, which exist in the system configurator, are still valid, because the IED configurator is not allowed to make changes which modify the already configured data flow. Especially, no names shall be changed, and no referenced parts of the data model shall be deleted. The general process is shown in Figure 4.



**Figure 4 - Modification process**

At system engineering start the IED capability (ICD) files are used by the system configurator to instantiate project specific IEDs as needed (1). The resulting SCD file is then imported by

the IED configurator for the final IED configuration (2). Any add-ons or adapted values could then be exported by means of the IID file, and thus brought back to the system configurator respective the next revision of the SCD file (3).

To allow detection of a modified file, within the IID file the IED owner should be set to the SCD header identification of the SCD, which was the input to IED engineering before.

## 5.5 Data exchange between projects

A complete secondary system can be split concerning the engineering responsibility into different parts. Examples are separate engineering of high voltage level and medium voltage level, of a transformer related part, or even of different substations exchanging data e.g. for line protection or interlocking. Such a system part with responsibility for all its contained IEDs is called a **project** in the scope of this document. To allow the engineering of online communication data flow between such projects, some interfacing data has to be exchanged between the projects, and the engineered interfaces have to be reimported to the concerned projects.

Note: from a statical point of view a *project* can be considered as a responsibility area. As the exchange of information between responsibility areas is done in processes, the term *project* is used here.

To facilitate this engineering data exchange, the following rules are set up:

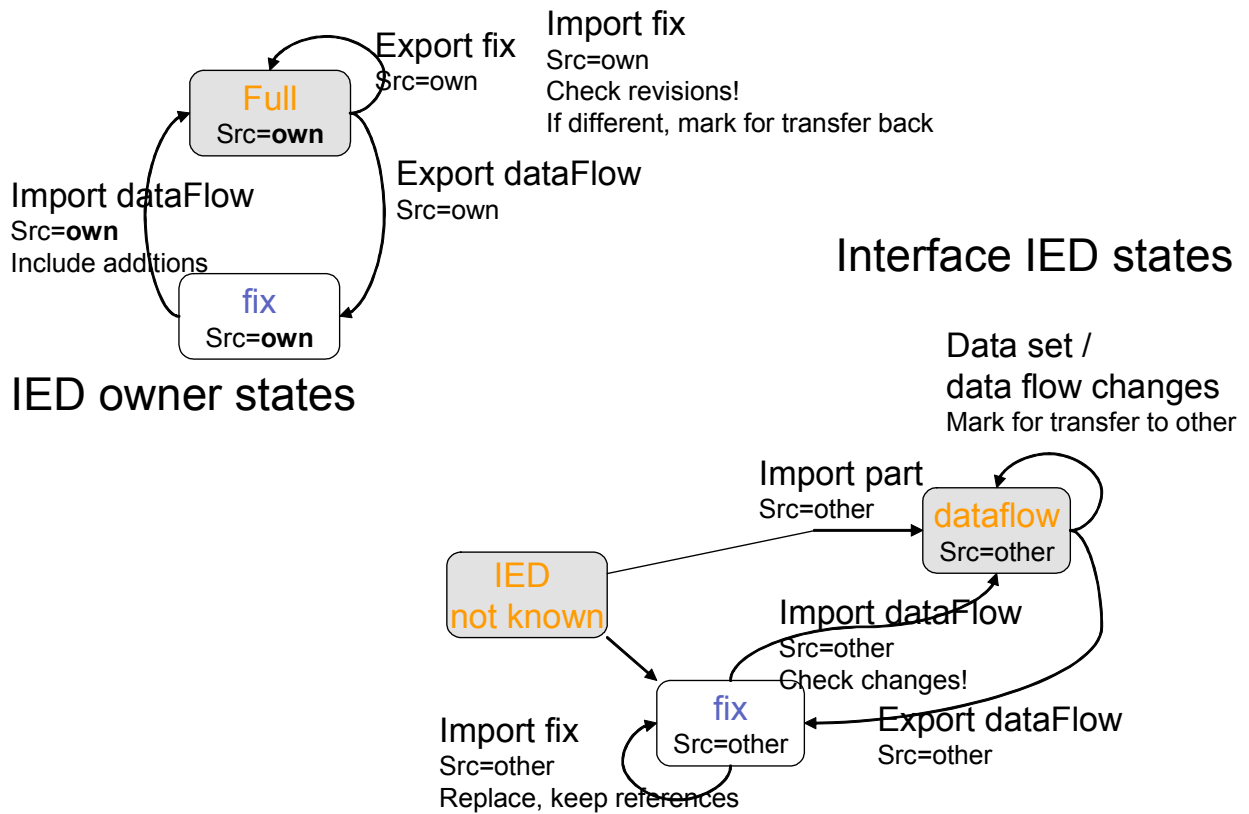
1. An IED always belongs to a project. Only people and tools belonging to the project are allowed to configure the IED, especially handling of data transfer between the project data base of the system tool and the IED tool. The owner project has full engineering rights on the IED.
2. A project can transfer the right to add data flow definitions to some of its IEDs to another project (IED check out). This has to be accompanied by a description of those parts of the IED which are allowed to be used and enhanced by the other project. This transfer of 'data flow' engineering rights blocks any modification of the exported IED in the owner project. Before this IED can be modified within the owner project, the 'data flow' engineering right has to be transferred back to it (IED check in), normally with some added data flow definitions.
3. To not loose already engineered references on such exported IED part, parts of referenced IEDs have also to be exported as fix IEDs. These are not allowed to be changed by the importing project, and must be reexported unchanged when the ready engineered IED is transferred back to the owner project.
4. Needed parts of the Substation section and communication section shall be exported. From the Substation section always full bays shall be exported, however containing only LN links to exported IEDs. The importing project is only allowed to add logical node links to the substation, or add a part of its own substation section. Further it is allowed to add addresses in the exiting subnetworks and own subnetworks to the communication section. It is up to the project engineer to assure, that entered own objects have unique names also in the exporting project, and that the entered addresses are unique within the full Subnetwork.

The rights transfer occurs by means of an SCL file, called a SED (System Exchange Description) file. This file also contains the transferred engineering rights (fix, or dataflow) and the IED engineering capabilities (e.g. number of data sets and control blocks allowed to be used by the receiving project).

The parts of an IED exported with *dataflow* right shall be frozen for engineering at the IED owner project, until the using project transfers it back by means of another SED file. The file transferred back shall have the same SCL Header identification as the originally exported file, however an increased revision index.

The Header identification of a SCD file is taken as project identification, and therefore it is always different to that of a SED, which identifies an engineering data transfer between exactly two projects. The SCD Header identification is used within a SED file to identify the owner project of an IED.

The transfer and handling of the *dataflow* and *fix* rights is defined in the state diagrams of the below Figure 5 for the project owning the IEDs as well as for the receiving project. Observe that the *full* engineering right always stays in the owner project.



**Figure 5 - Engineering right handling in projects**

The state diagram at the left side shows the internal states in the IED owner project, when exporting and importing owned IEDs via SED file. The right / lower state diagram shows the IED states of imported / exported IEDs at the receiving project, which is not owner of the IEDs.

The detailed engineering rights in terms of SCL elements is described in clause 10, after all elements have been introduced. It is recommended, that the temporary fix state of an IED with exported dataflow engineering right is also shown in the SCD file, if an SCD is produced from the project at that time.

If several (system) tools are used within the same project, it is the responsibility of the engineering people to use them in such a way, that the system description generatable as SCD file stays consistent. It is a project internal issue if in this case also the rights transfer mechanism is used within the project.

Note: The engineering right transfer can be considered as a checking out of the IED to a specific project, with later checking in again, after the data flow modifications have been done in the other project. Only one (other) project at a time is allowed to engineer the IED.

## 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 and optional DATA (here abbreviated DO, DATA Objects) as well as optional services, and are therefore not instantiable. Further it is allowed to add user defined DATA. In this document therefore 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.

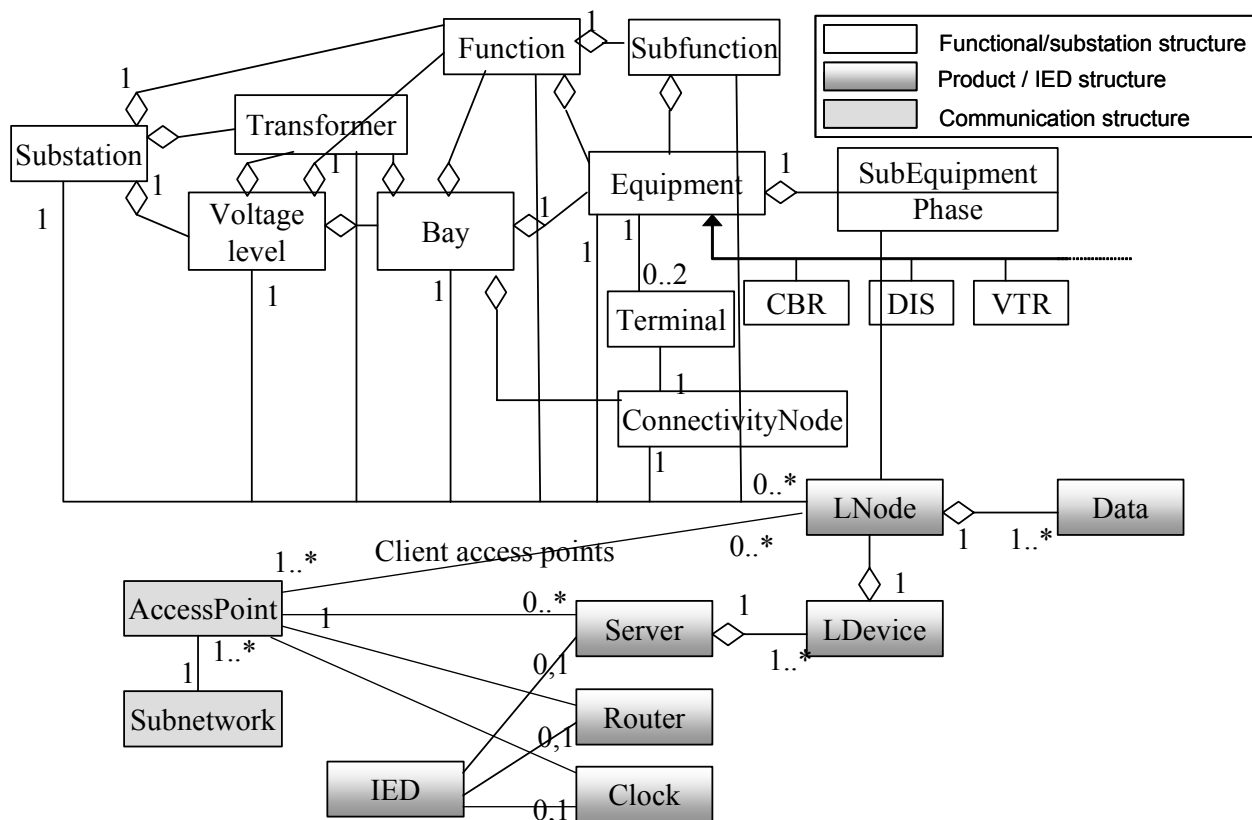
The UML object model is contained in Figure 6. Note that it is not complete in the modelling sense, i.e. it does not show any superclasses from which the used classes may be derived, no attributes etc. It restricts itself to those concrete object types that are used within a SCL instance file, in case 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 6 – 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 6, 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:

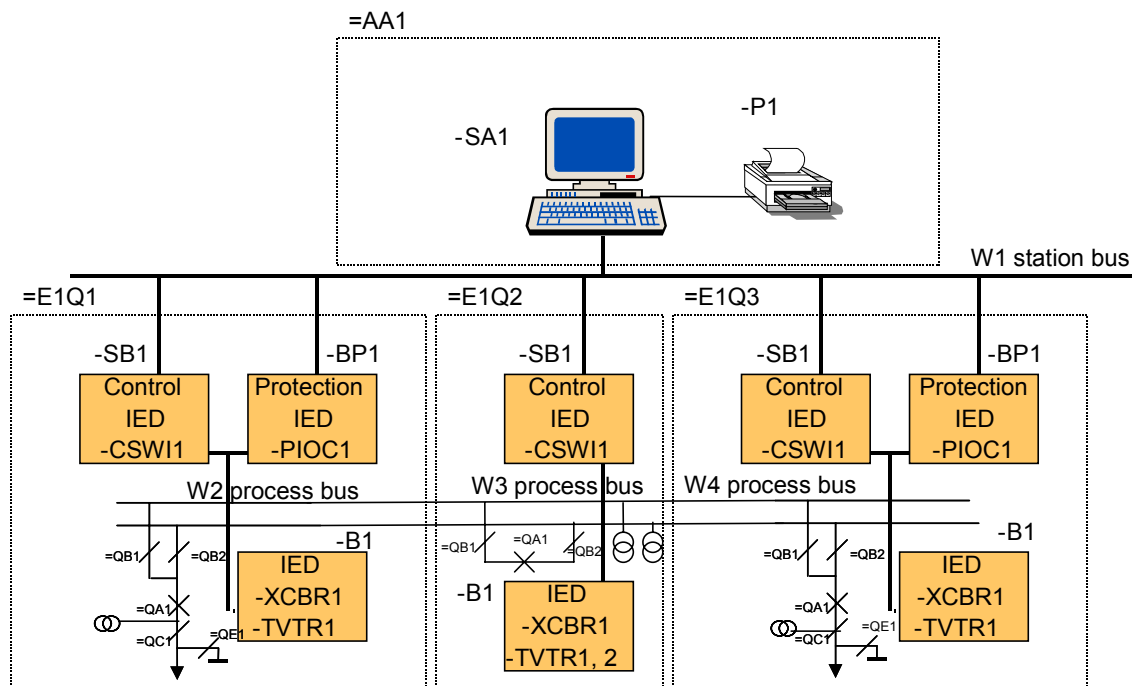
- A name is used as (a hierarchical part of) a technical key to designate the object. Each object within a hierarchy has an attribute *name*, which contains its identification within this level of the hierarchy. Technical keys are used in technical documentation for building and maintaining the system, 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 engineering time, and identifies a (functional) object at its hierarchy level to a human being. 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 (path name) unique within this level (see examples in 8.5).

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 7 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 could be seen in an SCL description in the name structure of the IED name. Only the transition at the logical node is modelled explicitly. Figure 7 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 7 – SA system configuration example**

## 6.2 The substation model

The substation model (upper part of Figure 6) 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 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) per equipment 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.
Function	allows additional functions at substation, voltage level or bay level, either independent from the basic switch yard functionality like fire fighting or building supervision, or as part of the switch yard like main 1 protection and main 2 protection.
Subfunction	a hierarchical subpart of the Function, e.g. earth fault protection as subpart of the main 1 function.

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 used for functional designations. If substructures of bays are needed, this can be introduced by appropriate structured 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.

NOTE 3 In the CIM model the bay level is optional, while in SCL it is mandatory. However, if the bay level structuring is not needed, a whole voltage level can be considered to be one bay. The only restriction here is that the SCL syntax demands at least one character as name on each level, so that in this case the voltage level name must have at least 2 characters, where then within the SCL substation structure the first character is taken as voltage level name, and the last character is taken as name for the one bay element.

### 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 <i>offered DOs</i> (server capability) are described in SCL. The <i>needed DOs</i> (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 6 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.

#### 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. Subnetworks may be connected by routers, however GSE messages as well as SV / SAV messages can not cross routers and can reach only IEDs within the same subnetwork. For accurate time synchronisation further each subnetwork should have an own (master) clock connected.

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, although some features allow to model it at least partly – see also 9.4.6.

This part of the standard introduces as additional IED functions:

- a *Router* function on an IED. An IED with a router function can be connected with two different access points to two different subnetworks and allow TCP based messages to reach IEDs within the other subnetwork.
- a *Clock* function to indicate where a subnetwork master clock is located.

Further the IED type SWITCH is reserved to model arbitrary switch based Ethernet networks, e.g. for IP address checking or modeling of the physical network. IEDs of type SWITCH typically consist only out of an access point to their IP subnetwork. IED type designations of the really used switches can be used instead, if known.

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.

Observe that the communication addresses defined for the access points within a subnetwork are access point addresses for building associations. The rules to derive communication addresses of the server internal elements are defined within the protocol mappings on the base of the IED data model as defined in IEC61850-7-x, e.g. within IEC 61850-8-1 for the MMS mapping.

## 6.5 Modelling of 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 SCSSM 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 normally not seen within an SCD file. Note however that SCL provides some optional means to describe physical connections at port / cable level including redundant ports.
- 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 all explicitly provided additional communication subnetworks are also modelled in the SCD file as indicated in Figure 7. Any coordination between redundant functions is done between the logical nodes which implement the function.

## 6.6 Data flow Modeling

Conceptually the IEC61850 data flow has servers or publishers as source, and logical nodes as clients respective subscribers. The real connections / associations however are built at MMS/TCP level, and these 'association channels' can be assigned to a client/subscriber IED, an access point, a logical device or – as in the 61850 base model, a logical node. If the channel is built by an IED, then all LNs hosted in this IED can use it as clients. It should be observed that these channels / associations are also the level of granularity on which access rights are checked, i.e. each association is seen as one user respective client with a certain role.

SCL allows to model the data flow at two levels. At the channel / association level the GOOSE or SMV subscribers are whole IEDs, respective the IED access point connected to the same SubNetwork as the server, while report clients are LN instances, like in the original IEC61850 client model. In case that also here more than one client LN share the same channel, it is

recommended to take the LLN0 as the client LN, because LLN0 represents a whole logical device.

At DATA level the data flow is modeled by a list of signals which shall be fed into (are input data of) a logical node. This can be modeled purely on SCL level, or, if the IED supports this, even at the IEDs LN data model by means of input DATA of CDC ORG (see IEC 61850-7-4). Also here it is often the case that the same incoming DATA shall be used by several logical node instances. In this case it is also recommended to map the input data into the LLN0 instead of mapping it twice to two different LN instances.

## 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 5.1 (see also Figure 1), there are at least five different purposes for SCL data exchange, and therefore five 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 defined in IEC 61850-7-3 / 7-4, or 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 5.1). This file describes the functional and engineering capabilities of an IED type. It shall contain exactly one IED section for the IED type 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 disconnecter implements the interlocking logic for a line disconnecter). 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 the IED configuration tool to the system configuration tool for a single IED preconfigured specifically for a project, e.g. to include a preconfigured instance file or IED instance value changes or data model modifications. In this case the IED has its project specific name, may be already project specific addresses, and a data model inclusive possibly some data set definitions preconfigured for the project. There might exist already a binding of IED LNs to the project specific single line diagram. This type of IED SCL file is typical for IEDs whose number of LN instances depends on the project specific single line diagram or on other IEDs available in the system, or it is used during IED modification process. It may contain data set and control block definitions, which must either be identical to those in the system tool in case that modifications after system engineering shall be transferred, or, in case of a first instantiation of this IED, can be taken as default or as preconfigured data. It may contain input sections without the referenced DATA sources. These shall be identical to that from a previously imported SCD file, however links to internal signals (intAddr values) may be added.

The file extension shall be .IID (Instantiated IED Description).

- Data exchange from a system specification tool to the system configuration tool. This file describes the single line diagram and functions of the substation and the required logical nodes. It shall contain a substation description section and may contain 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 a 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 5.1). This file contains all IEDs inclusive the needed *DataTypesTemplates*, 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 the communication related part of an instantiated IED within a project. The communication section contains the 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 (restricted view of source IEDs). If a compression method is applied, those according to RFC 1952 shall be preferred. Observe that in the general case more information than this has to be loaded onto an IED to have it completely configured, e.g. relation of internal signals to HW terminals, programs in form of IEC61131 or other code, or local control panel configuration information.

The file extension for the SCL part (if any) shall be .CID for Configured IED Description.

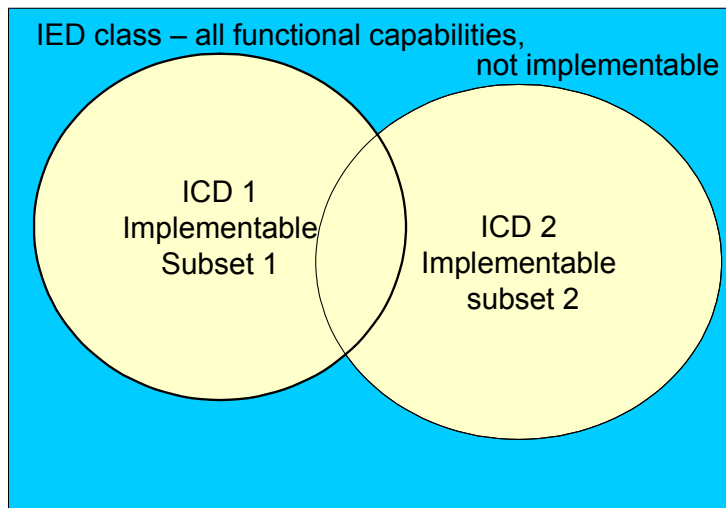
- Data exchange between system configuration tools of different projects. This file describes the interfaces of one project to be used by the other project, and at reimport the additionally engineered interface connections between the projects. It is a subset of a SCD file, containing the interfacing parts of the IEDs to which connections between the projects shall be engineered, and fix IEDs referenced by them to not loose already defined references. Therefore additionally to an SCD file it states at each IED the engineering rights and the owning project from the view of the using (importing) project.

The file extension shall be SED for System Exchange Description.

A more formal definition of most restrictions for the given parts is given in the XML schema syntax in Annex E. Observe however, that this formal definition is informative only and does not belong to the normative SCL language definition. Observe further that not all restrictions e.g. those 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 or client according to the IEC 61850 standard shall be accompanied by an ICD file respective by a tool capable to generate an ICD file, or a project specific IID file respective a tool capable to generate a project specific IID file for this IED, and shall be able to consume an SCD file respective 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 respective the IID file produced previously by the IED tool .

It shall be kept in mind, that for very flexible IED types there might exist several ICD files. In this case the manufacturers IED type can be seen as an IED class similar to logical node classes in IEC61850-7-4, which allows a lot of functionality to run on the IED HW, however not all at once. Each ICD file then is a runnable (implementable) subset of all possibilities of the IED class. Only in case that all available functions and function instances can run on the IED HW, there will exist only one ICD file. This issue is illustrated in Figure 8 for the most general case.



**Figure 8 – ICD files describing implementable IED types of a general IED class.**

## 8 The SCL language

### 8.1 Specification method

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

Its syntax definition is described as a 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 9 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 6. 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, 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 15. 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 describe 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 schema version, which is 3.0 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 9 gives an overview of how the SCL schema is structured.

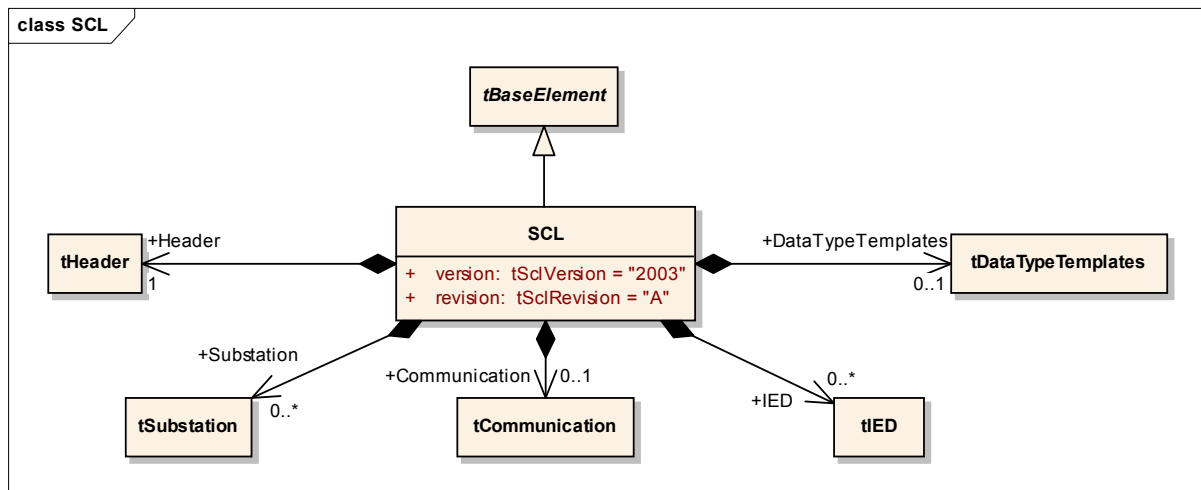


Figure 9 – UML diagram overview of SCL schema

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

In some cases, the data format of values is important. Wherever possible, the schema defines the data type and therefore also its coding (lexical presentation). But even in cases where this is not possible, the data type coding of XML Schema shall be used. If not explicitly expressed, all element values are XML Schema *strings*, and all attribute values are of the XML schema type *normalizedString*, i.e. they are not allowed to contain tab, carriage return and line feed characters. Further restrictions may be stated either in this part of IEC 61850 or in other parts of the IEC 61850 series, mostly IEC 61850-7-x, IEC 61850-8-x and IEC 61850-9-x. If any XML schema data type is used, it is referenced with the prefix *xs:*, for example *xs:decimal* for decimal number coding. For convenience, an overview about coding of the most types used in SCL is given in Table 45.

## 8.2 Language versions and compatibility

There are always some reasons why a defined language has to be changed, which leads to different language versions.

- Enhancements: adding new features; this has to be done to support new functionality, and leads to a new version indicated by the year of appearance, for this version it is 2007. To keep compatibility, the following enhancement rules have to be observed also for future compatible SCL versions. If for some reason they can no longer be observed, then a new, incompatible SCL name space has to be defined.
  - Adding of new optional attributes is allowed. If it makes sense, these shall have default values, whose meaning is identical to the missing of the attribute in older versions.
  - Adding of new elements is allowed at the end of existing type definitions.
  - To allow forward compatibility, any new element, whose understanding is essential for communication interoperability, must be marked with the *mustUnderstand* attribute (see later).
- Fixing errors; this is necessary if there exist faults or inconsistencies, or if interoperability problems arise due to unclear or wrong wording in the description or specification. This reason is mandatory and must be performed, even if it endangers compatibility. However, if there are choices, it should be done in the most backward compatible way. This leads to a schema revision, indicated by a revision index (letter) starting with A.

The following language changes are forbidden for a compatible language name space, because they lead to compatibility problems:

- Removing of old features. For backward compatibility they are still allowed in older language instances, but the usage in newer versions is deprecated. The deprecation is normally indicated by removing the feature from the schema of the new version. Nevertheless it is allowed in language instances coming from older versions, and must be accepted by a receiver.
- Changing of existing features, especially their semantics; this endangers compatibility, therefore it is forbidden. Instead 'old' features are deprecated (see above), and new ones added, which then replace the deprecated features.
- Changing of existing default values. This allows a receiver or processor to use default values of newer versions also for older language instances.

There is a clear separation between mandatory fixing of errors, which might lead to incompatibilities, and adding enhancements, which are done in a compatible way. To allow beneath backward compatibility also forward compatibility, the *may ignore* and *must understand* rules are introduced into the language definition. These allow to have different language versions for the same SCL language name space.

### 8.2.1 MustUnderstand rules

Elements, which a tool or an IED must understand to produce interoperable results, shall be declared as *mustUnderstand* and marked with the *mustUnderstand* attribute with value *true*, so that the tool processing the instance knows if it can ignore the element or not. All elements which the tool does not understand and which do not have the *mustUnderstand* property, can safely be ignored. The '*may ignore all*' strategy for elements (tags) is taken, i.e. ignore the element and all its contained contents. For attributes just the attribute not understood is ignored. This means especially, that there is no 'mustUnderstand' possibility for attributes, only for elements. Therefore adding of attributes to the language is done only as optional attributes with a defined default value in the newer version, which is backward compatible to 'not knowing this attribute'. For later compatibility it is good practice to use these default values from the schema by not explicitly writing them into the SCL instance. This is possible because once released default values are not changed in the schema, as long as the attribute itself is needed.

Observe that if attributes need to be understood, then a new element with *mustUnderstand* property holding these attributes can be introduced.

It is important to see that whether a tool can ignore something or not is also dependent on the purpose of the tool. When defining 'mustUnderstand' explicitly in SCL, then this always refers to the system configuration tool and the IED configuration tool as defined in clause 5 for the purpose of interoperable communication. Other applications for which SCL may be used can have other demands on 'mustUnderstand'.

Observe that although not formally defined, the *mustUnderstand* property is practically true for all defined elements from the 2003 SCL version in the *Communication* section, *IED* section (with exception of the IED capability element) and *DataTypeTemplate* section. The elements of the *Substation* section might need 'mustUnderstand' quality only for specification tools and application configuration tools, which are not mandatory and outside the scope of this part of IEC 61850.

From the meaning / scope, *mustUnderstand* always refers to the parent element. If a parent element (e.g. IED) has no *mustUnderstand* property, then it may be ignored by tools which do not need to know about it (e.g. about IEDs). However, if they have to know about IEDs, they must understand all elements within the IED element, which have *mustUnderstand* property, e.g. the *AccessPoint* element. In general, if an element is marked as *mustUnderstand*, then any tool which does not understand the element is not allowed to use the (known) parent element. User friendly tools will in any such case give a warning to the user.

### 8.2.2 SCL name space and versions

For all compatible versions of SCL the same name space as defined in subclause 8.3.5 is kept. For concrete verification of correct generation of an SCL instance according to its assigned version, the following is introduced.

The SCL element tag has a *version* attribute, which for backward compatibility is in general optional with default value 2003, and for instances assigned to the here defined version of SCL required with value 2007. This attribute indicates the SCL (schema) version according to which the SCL instance has been produced by means of the year of the released IS, i.e. its assigned version. Additionally any error fixing revisions within each version are indicated by the *revision* attribute, starting with A for the first released version revision. The version value for this version of SCL shall be 2007, and the revision value A. If error correcting revisions of this standard will follow before any new version of this standard, the first one will get the revision identification B, the next one C etc.

The special XML schema for this version 2007 is contained in the annex, and shall be used for all tests on SCL instances which claim to be produced according to this SCL version. From this follows automatically, that the SCL version attribute is mandatory required for all SCL instances containing elements or attributes introduced after 2003.

Note that for backward compatibility all tools have to accept SCL instances from older as well as newer versions, inclusive the features deprecated in the newer version(s) as far back as declared with the tool version. Therefore a tool input can not be verified against the schema of the version defined in this document, but at best against the schema version with which the input instance is produced. The general schema in Annex E gives a hint what shall be tolerated by a tool supporting the valid 2003 version as well as this 2007 version.

Naturally 'old' tools processing SCL instances from newer versions can not handle what they can not understand. The only problem which might arise here is if the SCL instance contains new elements with 'mustUnderstand' property, which is not known to tools/IEDs according to the first SCL version. Every tool/IED after this first version shall use the mustUnderstand property to decide if it can safely ignore an element, which is not understood, or if it has to stop processing with an appropriate error message. It is recommended to upgrade 'old' tools at least so that they can follow the mustUnderstand rule.

Further, tools understanding the new version should also read 'old' instances, if they are produced according to the rules defined here.

The SCL language version as defined here is related to the SCL XML schema version as defined in clause 8.1. However, not all XML schema versions will be released, and SCL language versions, even bug fix versions, will only be defined for released / published XML schema versions.

It might happen in the engineering process, that an IED tool supporting only an older SCL version writes back values (e.g. revision information) for the IED into a system SCL file of a newer version. In this case it shall keep all unknown (ignored) elements in the SCL file e.g. within other IEDs or within the Substation section, and has to reproduce them at the correct places, else it is not allowed to modify the file.

### 8.2.3 Incompatibilities to earlier versions

This 2007 version of SCL is backward compatible to all previous versions with the following exceptions:

- The authentication code 'week' has been corrected to 'weak' (error correction)
- The Private element's *type* attribute is required (error correction).
- The *sampleSynchronized* attribute of SMV options (agSmvOpts) is no longer allowed to be false (error correction in 9-2)
- The attribute value *FuncName* of the *nameStructure* attribute in the *Header* element is no longer supported. The *nameStructure* attribute shall be ignored by tools. Systems working with the 2003 functional naming must be modified appropriately to stay compatible, by either changing to IED based naming only, or allowing the use of the *IdName* attribute of *LDevice* (i.e. full change to this version of SCL).
- The introduction of the *mustUnderstand* attribute, currently only used at the *FCCB* element.
- The order where the Log element appears has changed: it shall appear directly before any control block definitions which belong only to LNO (like GOOSE control blocks).
- If the log control block does not reside in the same logical device as the log, the *IdInst* shall be stated explicitly.
- The access point name allows only alphanumeric characters and underscore (\_).

It is recommended to fix these issues together with the implementation of the `mustUnderstand` property also in 'old' tools, then they can correctly handle all future SCL versions.

### 8.3 SCL language extensions

#### 8.3.1 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 8.3.2 to 8.3.7 describe SCL extension possibilities.

#### 8.3.2 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.

#### 8.3.3 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.

#### 8.3.4 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.

#### 8.3.5 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. Therefore any SCL file shall have the SCL name space as default name space:

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

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 manufacturer's IED tool, if it is not contained within a *Private* section.

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.

### 8.3.6 Private data

For small extensions either by a manufacturer or for a specific project the *Private* elements can be used. The advantage of private elements 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 can not 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 a mandatory type attribute.
  </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="true">
    <xs:extension base="tAnyContentFromOtherNamespace">
      <xs:attribute name="type" type="xs:normalizedString" use="required"/>
      <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. The type attribute is required, to know who shall process this part.
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).

Observe that the data can be contained within the *Private* element, or in an external file referenced via the *source* attribute of the *Private* element. As a rule of thumb this second

option shall always be used if the amount of private data gets big in relation to the standardized part, e.g. above 1-2 kB.

### 8.3.7 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.

### 8.3.8 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.

### 8.3.9 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 – can be removed if not understood</ext:MyElement>
    <Private type="mytype" ext:hello="bla bla">This is my private element <ext:dummy>with sub-
elements</ext:dummy> and a privately defined attribute; must be reproduced at output</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.

## 8.4 General structure

An SCL – XML document starts with the XML *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"
      version="2007" revision="A">
```

```
<!-- 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). Anyhow, the *xsi:schemalocation* attribute is only needed if syntax verification against a specific schema shall be done.

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 9 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:attribute name="version" type="tScIVersion" use="required" fixed="2007"/>
        <xs:attribute name="revision" type="tScIRevision" use="required" fixed="A"/>
      </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 to contain elements and attributes from other name spaces. The elements derived from its sub-types *tUnNaming*, *tNaming*, and *tIDNaming* additionally inherit the *desc* attribute.

Observe further that the SCL element has the attributes *version* with value 2007 for this version of the SCL language, and *revision* with value A for this revision of the 2007 language version.

## 8.5 Object and signal designation

### 8.5.1 General

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", e.g. *iedName* as reference to an IED. 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 on purpose not specified further. Each tool shall preserve imported text data for export.

### 8.5.2 Object designations in an object hierarchy

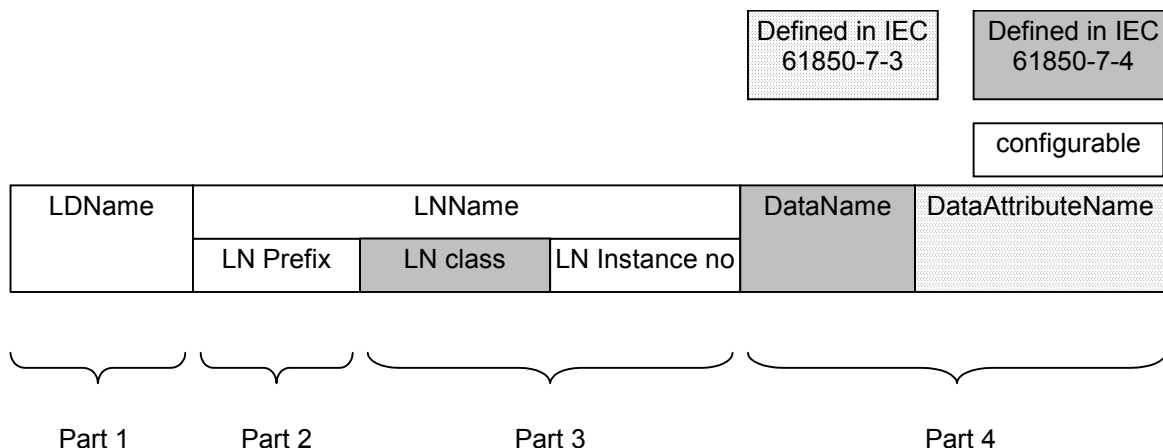
In case of the hierarchically structured objects of the substation structure and the product structure, both *name* and *desc* attributes for each object contain only that part which identifies the object within this level of the hierarchy. The full object reference is a pathname and consists of the 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. 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 IEC 61346 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.

### 8.5.3 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 10):

- 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 10 – Elements of the signal identification as defined in IEC 61850-7-2**

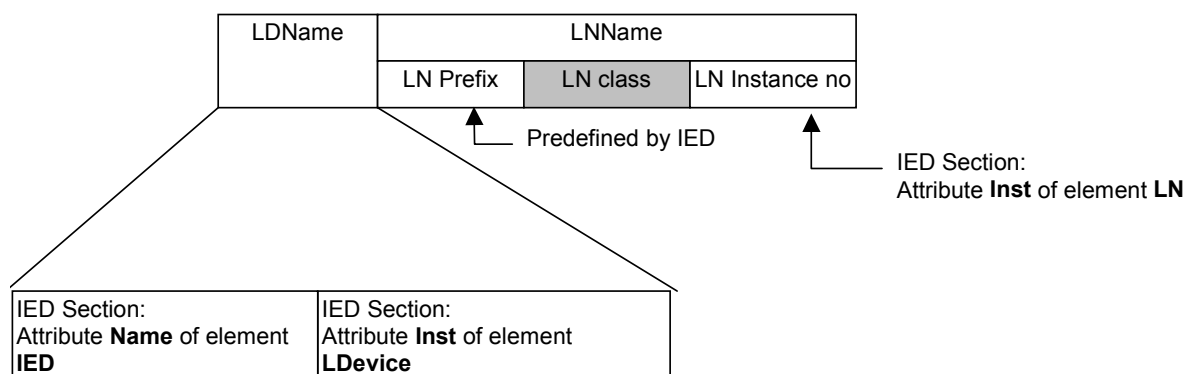
The name parts 2 and 3 in Figure 10 together form the LN name and distinguish different LN instances within the same LD of an IED. Both are not semantically standardized. A function related LN Prefix is preferably used during functional engineering, or to bind an instantiated LN on an IED to some process 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 *Data Type Templates* 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 10. The LN instance number and the prefix are defined in the IED section of the SCL.

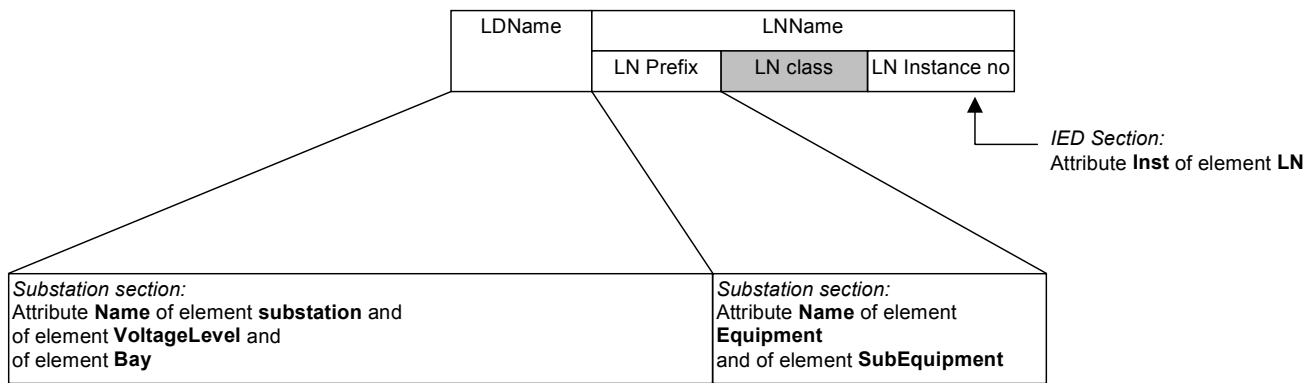
For name parts 1 and 2 in Figure 10 there exist several options, the most important two options are illustrated here (see also Figure 12 and **Error! Reference source not found.**).

- 1) **Product related naming:** As shown in Figure 11, part 1 in Figure 10 is the name of the IED in the IED (product) section, on which the LN is configured, concatenated with the IED relative LD Instance identification. Part 2 and 3 are as predefined within the IED **Error! Reference source not found.**



**Figure 11 – Elements of the signal name using product naming**

- 2) **Function related naming:** Function related naming at communication level is enabled by free setting of the LD name, and possibly free definition of the LD prefix. It is a decision of the IED manufacturer to allow one or both of this options by means of his tools. It has to be kept in mind, that these parts also have to obey special uniqueness restrictions, i.e. can not be completely freely used. The following usage could be possible: The LD name, part 1 in Figure 10, is the name of the switch yard function or function type, to which the LN relates. If it is a PrimaryDevice, you may use the name parts from substation name to bay name as part 1, and use the PrimaryDevice name (possibly followed by a sub equipment name) in part 2 (LN prefix). 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 10 stays empty, or can be used for the level where the LN is attached to. Observe, that according to part 7-2 the part 1 (LDName) must be unique within the subnetwork, i.e. it is not allowed to appear on two different IEDs connected to the same subnetwork. So, if you have a main1 protection IED and a main2 protection IED in the same bay E1Q1, each with exactly one logical device for protection, and you use functional naming via the *ldName* attribute, then the LD Name could e.g. be E1Q1F1 for main 1, and E1Q1F2 for main 2 (F stands for protection functions in IEC 61346).



**Figure 12 – Possible elements of the signal name using functional naming**

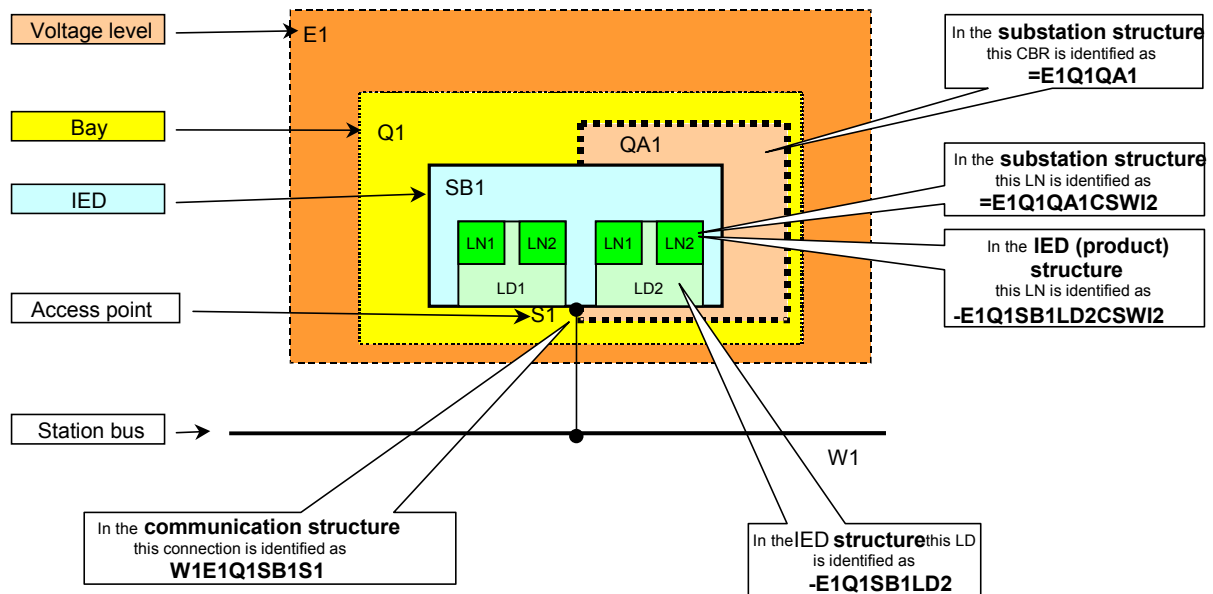
The SCL language allows both options, even separate for different IEDs. The mandatory option is the product related naming. If function oriented naming is needed, the (optional) function oriented LD name has to be explicitly specified for each logical device. It is recommended to use the LN instance 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 features might be restricted by the IED manufacturer in case that an IED has a fixed prefix and LN instance number, i.e. does not allow to change this for a certain LN instance later on. In this case function related naming can be chosen only at LD level. Observe that also the LD inst name part might be fix for a certain IED type, because it serves as manufacturer identification of the logical device on this IED type. The IED name however and the (function oriented) LD name, if supported at all, shall be freely choosable by a system integrator. In any case, also for product related naming, the meaning of a LN in the context of the switch yard can be established via the LN link from the substation section to the IEDs.

**8.5.4 Signal identifications usable by applications**

The communication related names, even if function oriented, depend on manufacturer supplied engineering capabilities as well as the concrete distribution of logical nodes on the IEDs. Applications needing a functional view independent from this should use a signal identification based on the Substation structure names down to the LN class, and then followed by the semantically completely standardized DATA and attribute names. A switch position could then be identified by the path name <substation name (AA1)><voltage level name (J1)><bay name (Q1)><Equipment name (QB1)>CSWI.Pos, an earth fault protection function in main 1 e.g. by the path name <substation name (AA1)><voltage level name (J1)><bay name (Q1)><function name (Main1)><Subfunction name (EF1)>PTOC.Op. The SCL language allows this kind of application related naming in parallel to the communication related naming, and a complete SCD file might serve a data base to translate from one to the other.

**8.5.5 Naming example**

Figure 13 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 13 shows the resulting references within different structures, and the resulting LN reference for communication.



**Figure 13 – 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, i.e. the LD2 LDName would be the bay name E1Q1, then the LN reference according to IEC 61850-7-2 would be E1Q1/QA1CSWI2. If the references were taken from the product structure, it would be E1Q1SB1LD2/CSWI2. Observe that the whole name in each case shall be unique within the subnetwork, which is the case for both names above. However, in the case of the functional name, the LD reference E1Q1 alone is **not necessarily unique within the subnetwork** (only within the IED). It is the responsibility of the project engineer to assure that there is no other IED with LDName E1Q1 – which restricts the system architecture with functional naming to one IED per bay. The application level functional name E1Q1QA1CSWI however is again unique, and independent from communication level functional or IED related naming.

## 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 14 gives an overview on its structure.

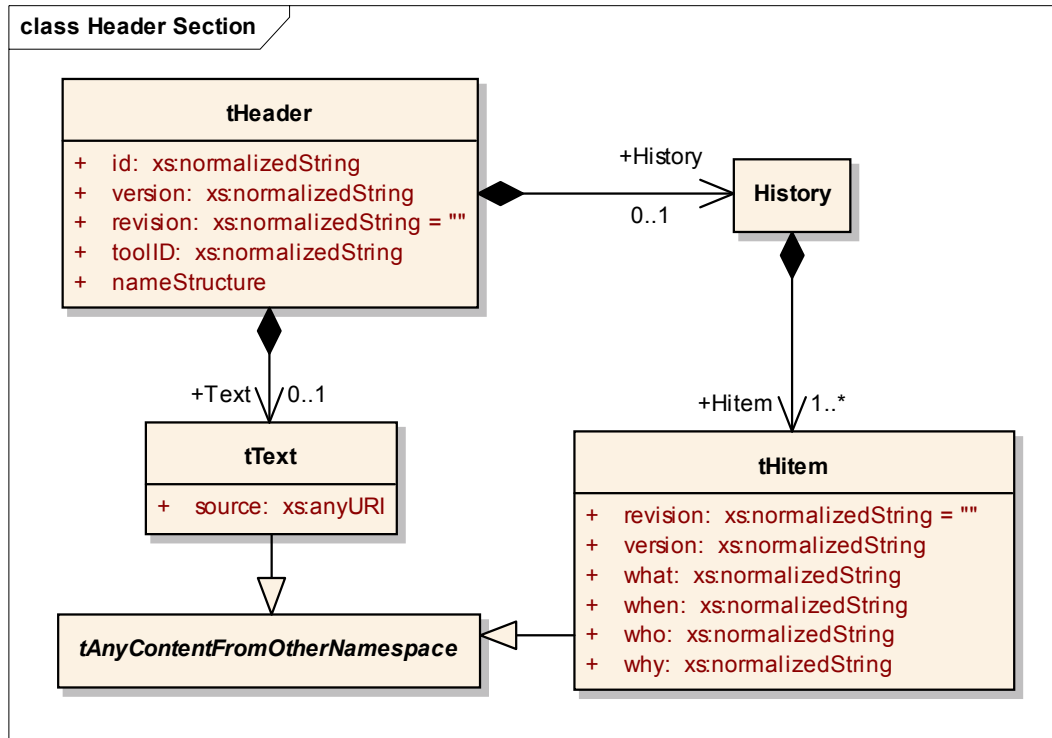


Figure 14 – UML diagram of Header section

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="optional" default="IEDName">
    <xs:simpleType>
      <xs:restriction base="xs:Name">
        <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 (can be empty)
version	The project specific version of this SCL configuration file (can be empty)
revision	The project specific 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 provided optional only for backward compatibility with previous SCL schema version. If given at all, only the IEDName value is allowed.
---------------	---

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 header example without history:

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

## 9.2 Substation description

### 9.2.1 General

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 of Figure 15 gives an overview on the substation section:

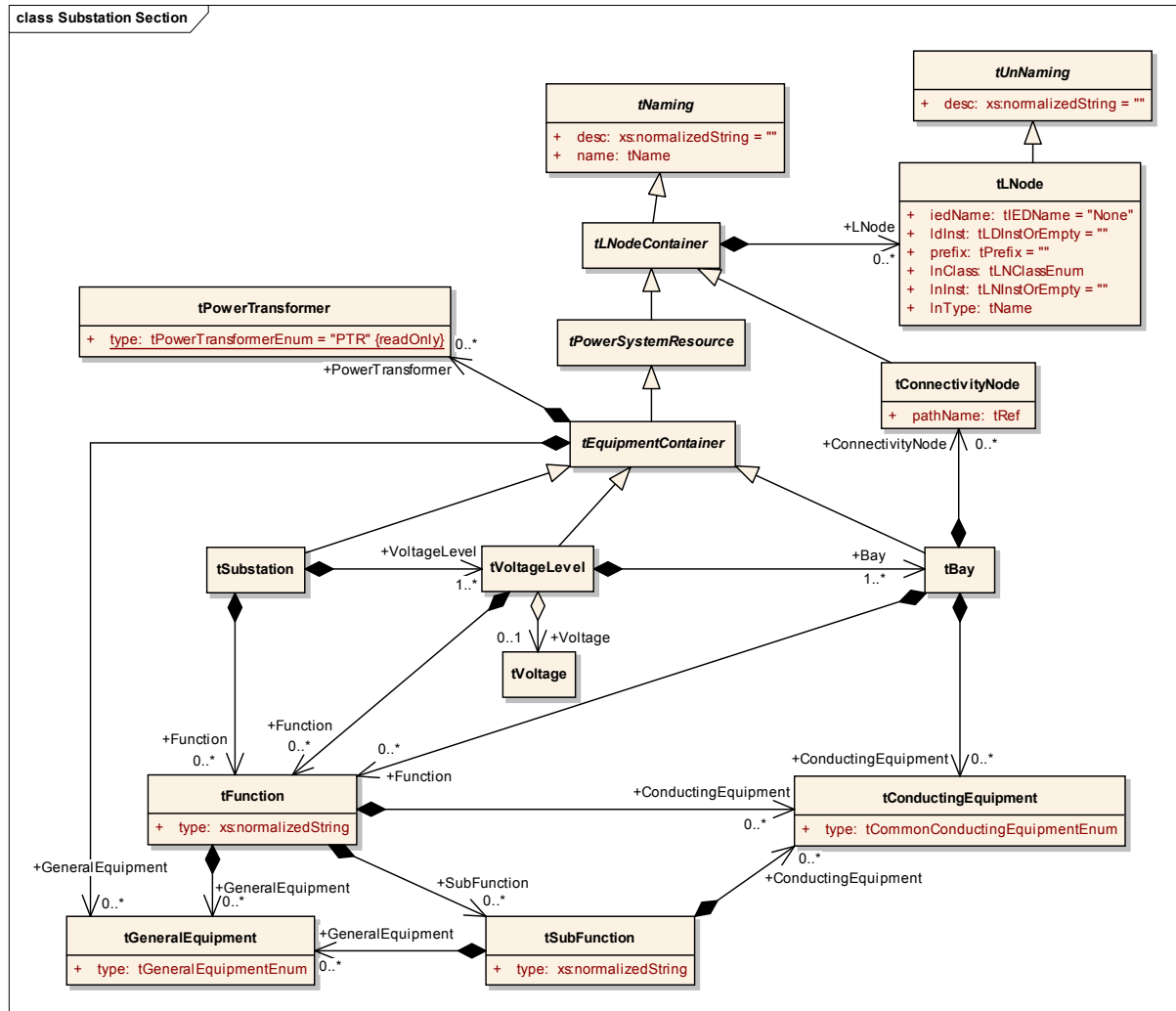


Figure 15 – UML diagram of Substation section

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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <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>

```

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="uniqueSubFunctionInFunctionVL">
            <xs:selector xpath="/scl:SubFunction"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInFunctionVL">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

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

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

- Within a *Substation*, there cannot be two *VoltageLevel* elements with the same *name*.
- Within a *Substation*, there cannot be two direct *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 *InInst*, *InClass*, *iedName*, *IdInst*, and *prefix*.

- Further, in order to avoid any ambiguities, within a *Substation* there cannot be two direct child elements with the same *name*.
- In general, at each hierarchy level within the substation section all names shall be unique, leading to unique object references (path names) of all objects defined by the substation naming hierarchy.

### 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.

### 9.2.2 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, and may contain *Function* elements.

```
<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: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>
```

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 and function name shall be unique within a voltage level.

### 9.2.3 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 conducting equipment and power transformers within a single line diagram, and *Function* elements, e.g. for different protection functions.

```

<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:element name="Function" type="tFunction" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueSubFunctionInFunction">
            <xs:selector xpath="/scl:SubFunction"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInFunction">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The *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 Annex A):

- Within a *Bay*, there cannot be two direct child elements *PowerTransformer* with the same *name*.
- Within a *Bay*, there cannot be two direct child elements *ConductingEquipment* with the same *name*.
- Within a *Bay*, there cannot be two direct child elements *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 9.2.8.

NOTE 3 If no bays are needed within a voltage level, then the whole voltage level can be modelled as just one bay. It has only to be kept in mind, that this virtual bay needs a name of at least one character length.

### 9.2.4 Power equipment

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

The UML diagram given in Figure 16 gives an overview about the equipment inheritance relations.

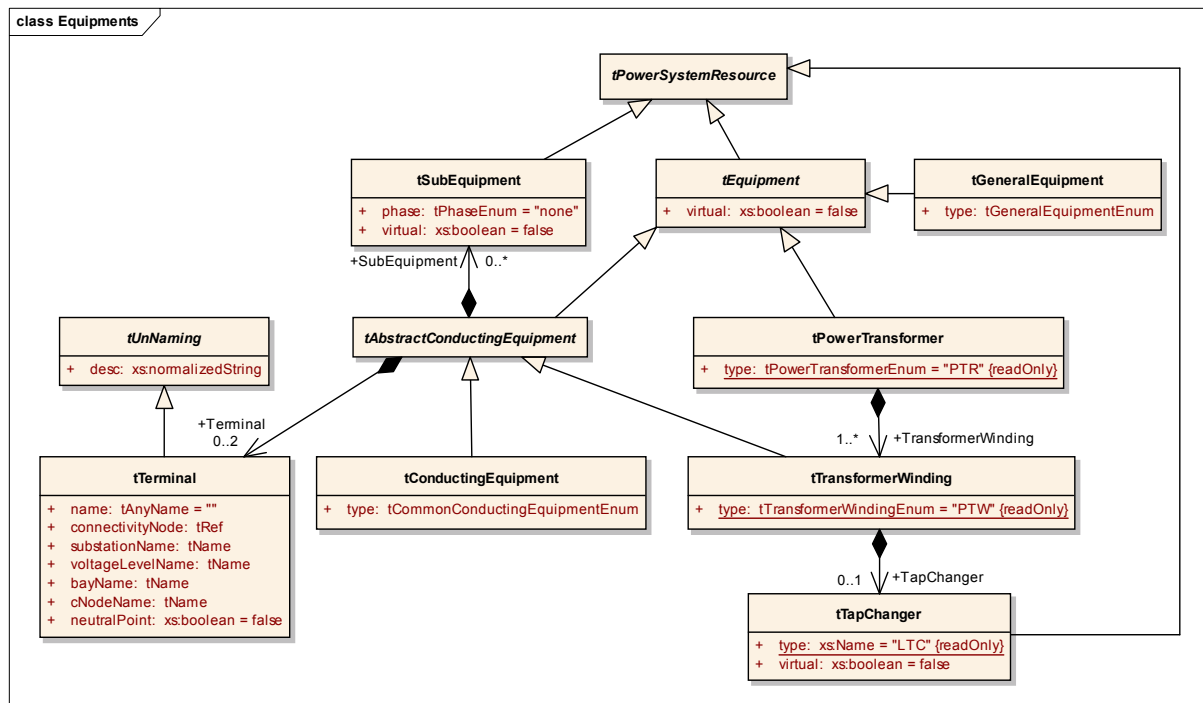


Figure 16 – UML diagram for equipment type inheritance and relations

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

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

**Table 5 – Primary apparatus device type codes**

Type code	Meaning	Number of terminals (connections to different connectivity nodes)
CBR	Circuit Breaker	2
DIS	Disconnect or earthing switch	2
VTR	Voltage Transformer	1
CTR	Current Transformer	2
PTW	Power Transformer Winding	1/2
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
FAN	Fan	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
RES	Neutral resistor	2
RRC	Rotating reactive component	1
SAR	Surge arrester	1
SCR	Semiconductor controlled rectifier	2
SMC	Synchronous Machine	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.

Observe that the second terminal for a power transformer is only foreseen for a neutral point connection terminal, to which e.g. (one phase) earthing switches can be connected.

A terminal definition contains the reference to a connectivity node to which the equipment is connected (ConnectivityNode in the model of Figure 6), 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:attribute name="neutralPoint" type="xs:BOOLEAN" use="optional" default="false" />
    </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
neutralPoint	If true, this terminal connects to a neutral (star) point of all power transformer windings. Default value is false.

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 is 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.

### 9.2.5 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 ConductingEquipment.

```

<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 subEquipment relative to the device
phase	The phase to which the subEquipment belongs. The following phase values are allowed: <i>A</i> , <i>B</i> , <i>C</i> , <i>N</i> (neutral), <i>all</i> (meaning all three phases), <i>none</i> (default, meaning not phase related). The following additional values are only allowed, if the ConductingEquipment above has type VTR: <i>AB</i> , <i>BC</i> , <i>CA</i> , meaning a VT connected in between the appropriate phases.
virtual	Set to <i>true</i> , if the subEquipment (for example phase CT) does not exist in reality, but its values are just calculated. Optional, default is <i>false</i>

### 9.2.6 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="iedName" type="tIEDName" use="optional" default="None"/>
      <xs:attribute name="IdInst" type="tLDInstOrEmpty" use="optional" default=""/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
      <xs:attribute name="InClass" type="tLNClassEnum" use="required"/>
      <xs:attribute name="InInst" type="tLNInstOrEmpty" use="optional" default=""/>
      <xs:attribute name="InType" 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 *InType* may refer to a logical node type definition (9.5.2), 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 *InType* 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
InInst	The LN instance identification. Can only be missing for InClass=LLN0, meaning as value here the empty string.
InClass	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).

IdInst	The LD instance on the IED which contains the LN within a specification (SSD file), where <i>iedName=None</i> , this shall result in unique LN instance identification, i.e. may contain the LD name
prefix	The LN prefix used in the IED (if needed; default, if not specified, is the empty string). Can be used for more detailed function specification than possible by LN class alone, if the LN is not allocated to an IED
InType	The logical node type definition containing more detailed functional specification. Might be missing, if the LN is allocated to an IED.

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

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

### Restrictions

- A logical node can only be referenced once within all substation sections.
- Therefore, the combination of *iedName*, *IdInst*, *prefix*, *InClass* and *InInst* shall be unique within all substation sections.
- The naming conventions for all these name parts shall be followed, even if used within a specification.

NOTE For specifications, where *iedName="None"* everywhere, the combination of the other attributes must be unique within the same level. This means e.g. that the prefix or *InInst* should be different if several LNs with identical *InClass* are used within the same substation part (i.e. same bay). This should also be the case for 'real' IEDs, if functional naming shall be used. This is NOT checked by the SCL schema, therefore it is the responsibility of the project engineer or system tool, if functional naming shall be used additionally to product related naming.

### 9.2.7 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. It is possible to have *ConductingEquipment* within these elements to model relations between electrical and non electrical parts. The *ConductingEquipment* then might contain (electrical) connection definitions to the electrical part.

```

<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:element name="ConductingEquipment" type="tConductingEquipment" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
    </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:element name="ConductingEquipment" type="scl:tConductingEquipment" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
  </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>

```

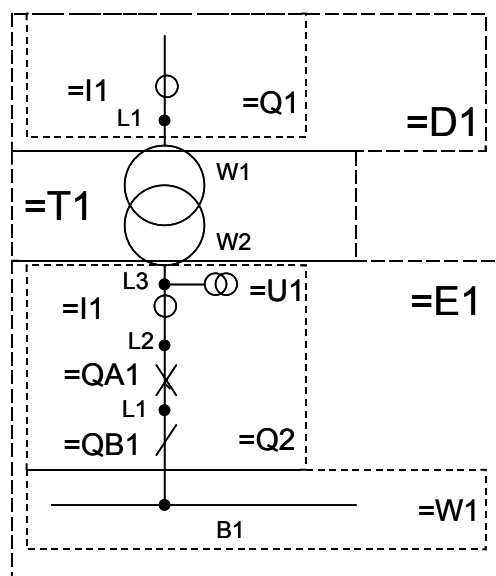
From the conducting equipment type list (Table 5) this is AXN, BAT, MOT, FAN, additionally to the codes defined in Table 9. Furthermore, all private codes (containing only capital letters, starting with "E") can be used. Other parts of this standard will define more type codes.

**Table 9 - General Equipment codes from IEC 61850-7-4**

Type code	Meaning
FIL	Filters
PMP	Pumps
VLV	Valves

### 9.2.8 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 .



### Figure 17 - Substation section example

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 L1, as well as voltage transformer U1 at connectivity node L3, and current transformer I1 between the connectivity nodes L3 and L2. The connectivity node within the same bay is explicitly defined. A logical node of type CSWI controls each switch, and the LN CILO handles the interlocking. No association to IEDs is defined, as this is a functional specification only, so the *iedName* is per default None. In addition, the possibility of defining more details by *InType* references has not been used here.

```
<?xml version="1.0"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2007" revision="A">
  <Header id="SSD Example" />
  <Substation name="baden220_132">
    <PowerTransformer name="T1" type="PTR">
      <LNode InInst="1" InClass="PDIF" IdInst="F1"/>
      <LNode InInst="1" InClass="TCTR" IdInst="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 InInst="1" InClass="PDIS" IdInst="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 InInst="1" InClass="CILO" IdInst="C1" />
          <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 InInst="2" InClass="CSWI" IdInst="C1"/>
          <LNode InInst="2" InClass="CILO" IdInst="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 the 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:

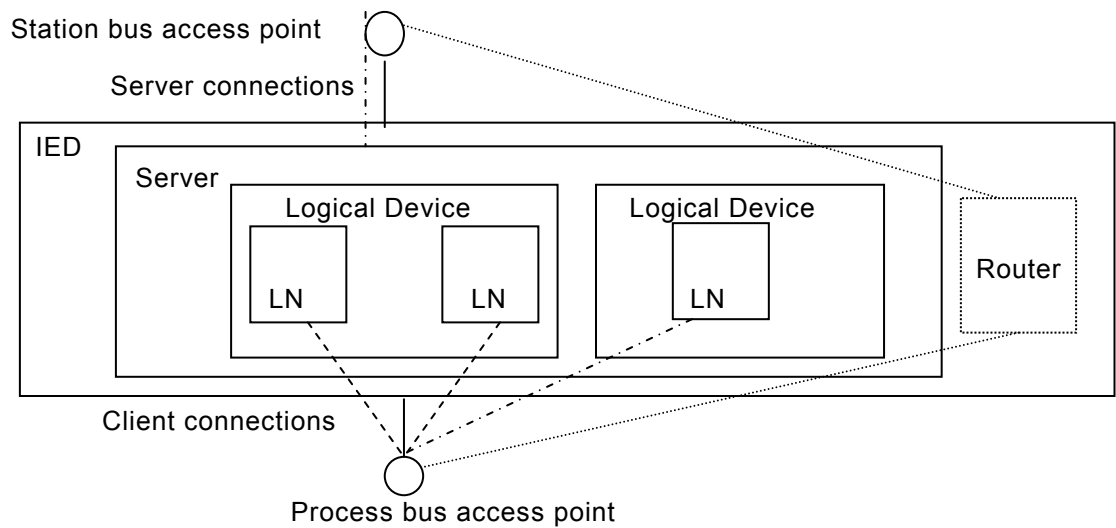
- time synchronisation messages,
- GSE messages,
- sampled analog measurement values.

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, an IED without a server may 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 18 gives a typical example of an IED connected to station bus and process bus.



**Figure 18 – 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.

Figure 19 to Figure 21 give an overview of the IED related schema part in the form of UML diagrams.

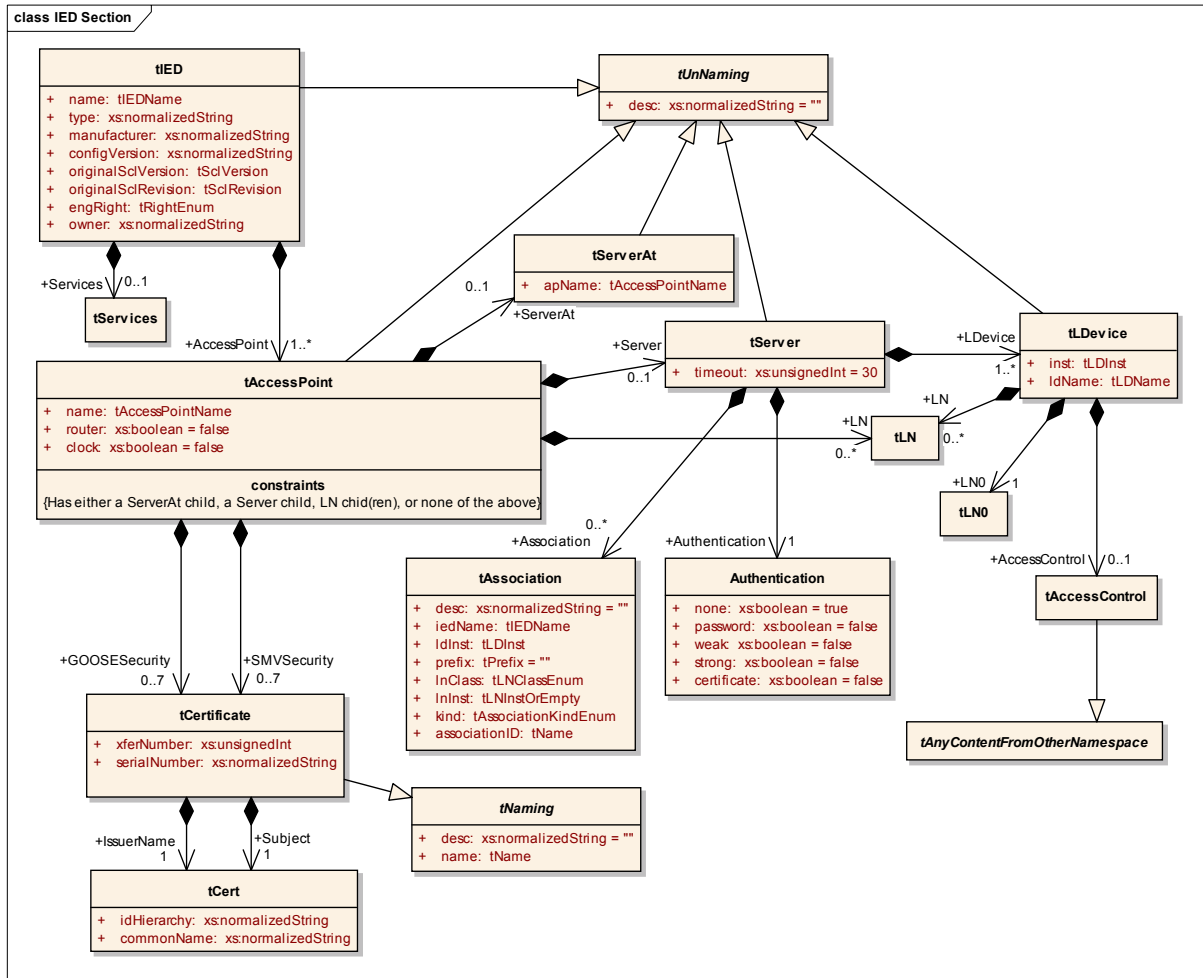


Figure 19 – UML description of IED related schema part – base

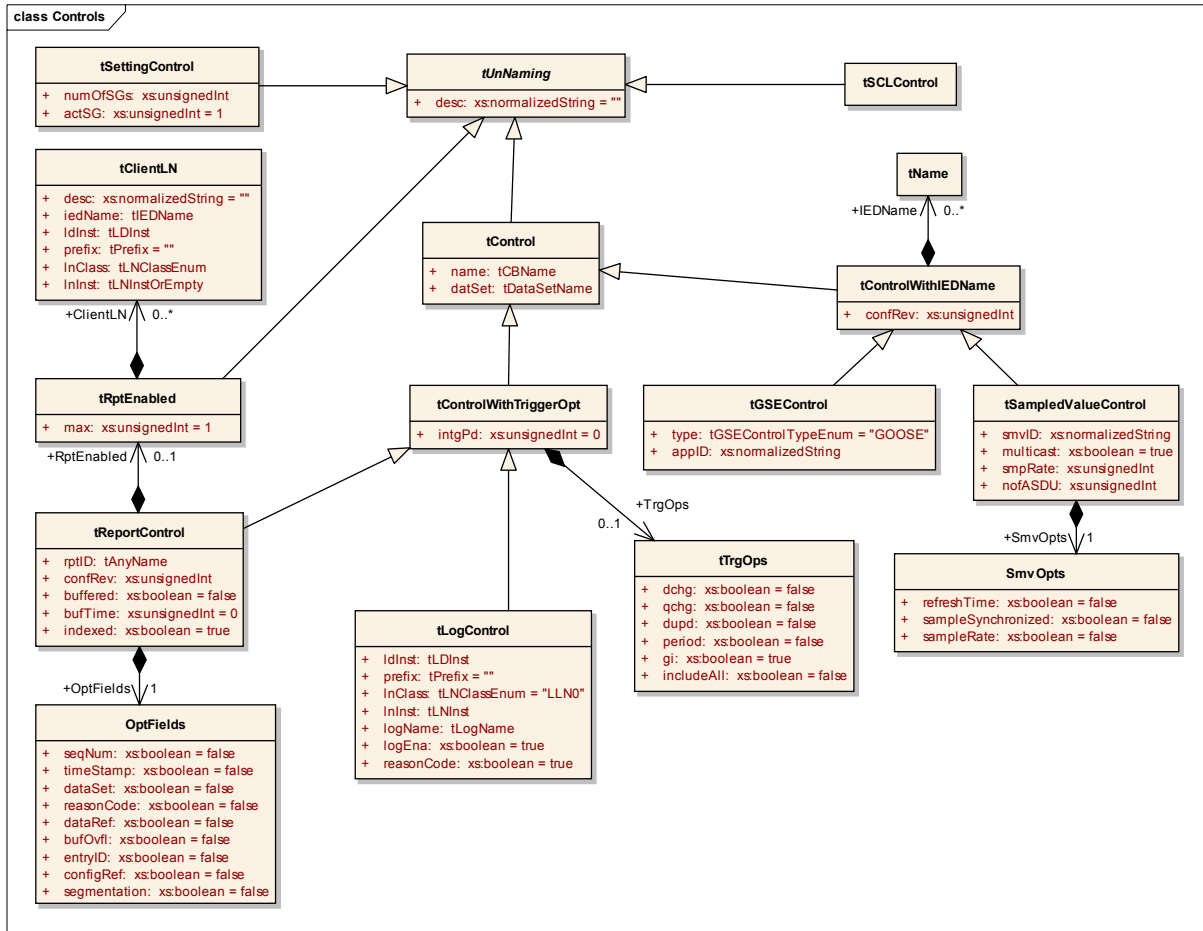


Figure 20 – UML description of IED related schema part for Control blocks

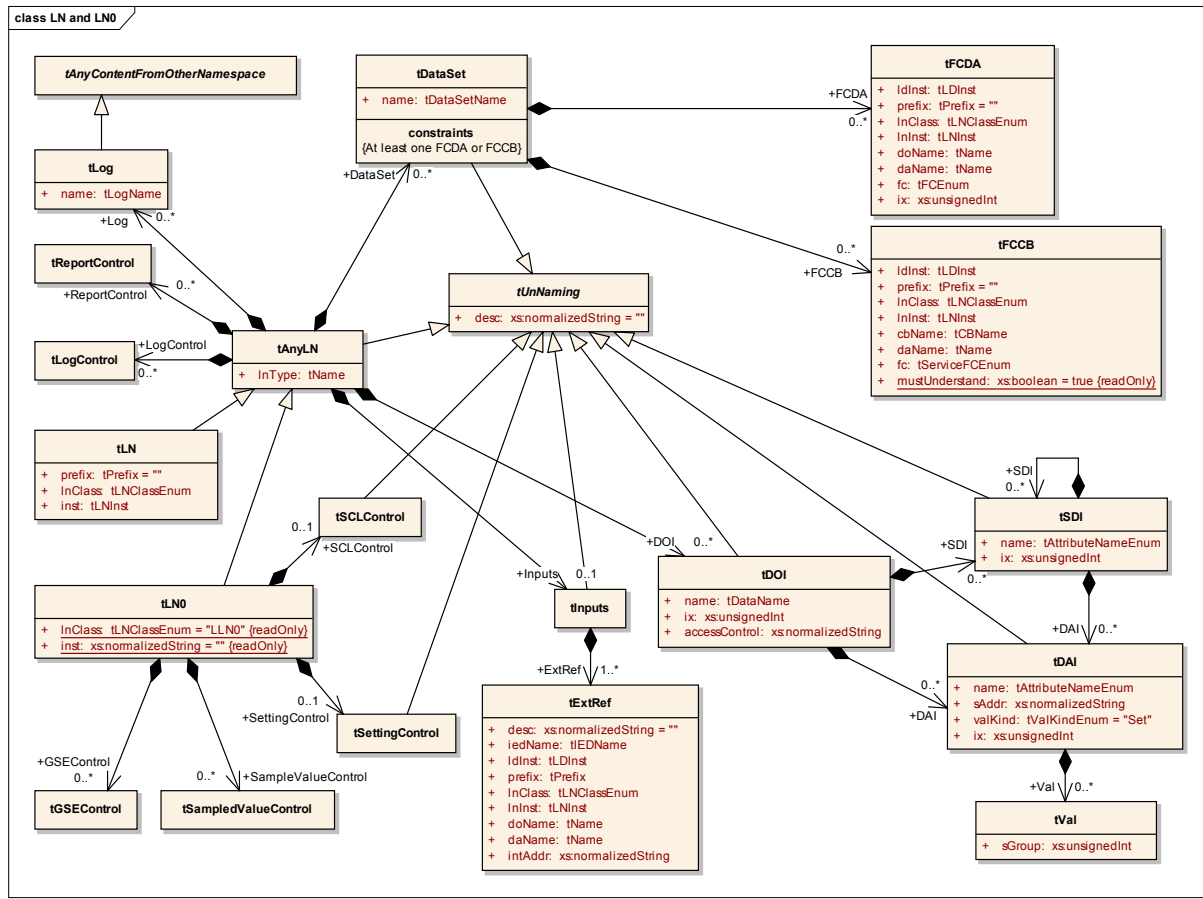


Figure 21 – UML description of IED related schema part – LN definition

### 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="@InClass"/>
            <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:attribute name="originalSclVersion" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="originalSclRevision" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="engRight" type="tRightEnum" use="optional" default="full"/>
      <xs:attribute name="owner" type="xs:normalizedString" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
  
```

The attributes of the IED element are defined in Table 10.

**Table 10 – 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
originalScIVersion	The original SCL schema version of the IEDs ICD file; optional
originalScIRevision	The original SCL schema revision of the IEDs ICD file; optional, shall be supplied if originalScIVersion is supplied.
engRight	The engineering right transferred by a SED file (only fix, dataflow), or the current state in an SCD file. Values are full, dataflow, fix, the default is full.
owner	The owner project of this IED, i.e. the Header id of that SCD file of that project which has the right to use the IED tool for this IED. The default is the Header id of the SCD file containing the IED.

The IED *configVersion* above only identifies the IED basic configuration (IED type capabilities as defined/delivered by the manufacturer e.g. after preengineering of a flexibly engineerable IED type), and not its individual configuration after instantiation into a project. The version of the project specific IED is a parameter of the IED instance, or of its logical nodes. It shall be contained in a SCL file as an attribute value of the attribute *LLNO.NamPIt.configRev*.

The *originalScIVersion* states the SCL version of the ICD file of this IED, and *originalScIRevision* the SCL revision of the ICD file. It is set by the IED tool when creating the ICD file. In case that it is missing, this original version is not known.

The IED contains a Service capability list, and access point definitions.

### Restrictions

- The IED name shall be unique within the SCL file.
- The length of the IED Name shall be at least one, at maximum 32 characters. It starts with an alpha character, and contains only alphanumeric characters and the underscore character. Note that there might be more restrictions in other parts of this standard, or due to usage of this name at engineering time.
- The IED name for an IED template shall be TEMPLATE.

The general *IED* element (of type *tIED*), which is contained within 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 *IdName* attribute of each *LogControl* may refer 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="tServiceWithOptionalMax" 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="DataSetDirectory" type="tServiceYesNo" minOccurs="0"/>
<xs:element name="ConfDataSet" type="tServiceForConfDataSet" 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="tServiceConfReportControl" 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="SMV" type="tServiceWithMax" minOccurs="0"/>
<xs:element name="FileHandling" type="tServiceYesNo" minOccurs="0"/>
<xs:element name="ConfLNs" type="tConfLNs" minOccurs="0"/>
<xs:element name="ClientServices" type="tClientServices" minOccurs="0"/>
<xs:element name="ConfLdName" type="tServiceYesNo" 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. For the meaning of the services, refer to IEC 61850-7-2.

The list of service capabilities and setting elements and attributes are described in Table 11.

**Table 11 – List of service capabilities and setting elements and attributes**

Service capability	Description
ClientServices	Indicates which general service classes this IED can use as a client: <i>goose</i> , <i>gsse</i> , sampled values ( <i>sv</i> ), unbuffered reporting ( <i>unbufReport</i> ), buffered reporting ( <i>bufReport</i> ), reading logs ( <i>readLog</i> ). Default (missing element): supported client services not known (except possibly from GOOSE/GSSE elements) – look into PICS. Required for 2007 version. A pure client shall set at least one of the options to true.  Additionally the attribute <i>fccb</i> indicates that the client supports data sets with FCCB definitions, i.e. references to control block attributes.  Default for a missing attribute: false.
DynAssociation	All services for dynamic building of associations. These are capabilities without attributes.  The <i>max</i> attribute indicates the maximum guaranteed number of dynamic associations which are possible.
SettingGroups: SGEdit ConfSG	Setting group services belong to the setting group control block. If this control block is available, then the setting group service <i>SelectActiveSG</i> for activating a setting group is also available. The capability of online editing (IEC 61850-7-2 services <i>SelectEditSG</i> , <i>ConfirmEditSGValues</i> , <i>SetSGValues</i> ) is decided with the <i>SGEdit</i> element. The capability to configure the (number of) setting groups by SCL can be also available ( <i>ConfSG</i> ). These are options without attributes.
GetDirectory	Services for reading the contents of a server, i.e. 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 <i>GetServerDirectory</i> , <i>GetLogicalDeviceDirectory</i> , <i>GetLogicalNodeDirectory</i>

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	<p>If ConfDataSet is not specified, then the default value of its <i>max</i> attribute is equal to the number of preconfigured data sets, and they may only 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.</p> <p>The attribute meaning is:</p> <p><i>max</i> – the maximum number of data sets  <i>maxAttributes</i> – the maximum number of attributes allowed in a data set (an FCDA can contain several attributes)  <i>modify</i> – TRUE means that preconfigured data sets may be modified; default: false  <i>fccb</i> – TRUE means that a data set may contain the FCCB element; default: false</p>
DynDataSet	<p>Services to dynamically create and delete data sets. Refers to IEC 61850-7-2 services CreateDataSet and DeleteDataSet.</p> <p>The attribute meaning is:</p> <p><i>max</i> – the maximum number of dynamically creatable data sets (including eventually predefined data sets)  <i>maxAttributes</i> – the maximum number of attributes allowed in a data set (an FCDA can contain several attributes)</p>
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 <i>ctlModel</i> attribute. It is a service without attributes.
ConfReportControl	<p>Capability of static (by configuration via SCL) creation of report control blocks.</p> <p>The attribute's meaning is:</p> <p><i>max</i> – the maximum number of instantiable report control blocks. If this is equal to the number of preconfigured instances, then no new instances can be created.</p> <p><i>bufMode</i> – unbuffered, buffered, both; the buffer mode allowed to configure for new control block types.</p> <p><i>bufConf</i> – boolean. TRUE means, the <i>buffered</i> attribute of preconfigured report control blocks can be changed via SCL.</p>
GetCBValues	Read values of control blocks. It is a service without attributes.
ConfLogControl	<p>Capability of static (by configuration via SCL) creation of log control blocks.</p> <p>The attribute's meaning is:</p> <p><i>max</i> – maximum number of instantiable log control blocks.</p>
ReportSettings	<p>The report control block attributes for which online setting is possible with services SetURCBValues respective SetBRCBValues:</p> <p>The attribute's meaning is:</p> <p><i>cbName</i> – control block name  <i>datSet</i> – data set reference  <i>rptID</i> – report identifier  <i>optFields</i> – optional fields to include in report  <i>bufTime</i> – buffer time  <i>trgOps</i> – trigger options enable  <i>intgPd</i> – integrity period</p>

LogSettings	<p>The log control block attributes for which online setting is possible with service SetLCBValues:</p> <p>The attribute's meaning is:</p> <p>cbName – control block name</p> <p>datSet – data set reference</p> <p>logEna – log enable</p> <p>trgOps – trigger options</p> <p>intgPd – integrity period</p>
GSESettings	<p>The GSE control block attributes for which online setting is possible with service SetGsCBValues respective SetGoCBValues:</p> <p>The attribute's meaning is:</p> <p>cbName – control block name</p> <p>datSet – data set reference</p> <p>applID – application identifier</p> <p>dataLabel – value for the object reference if the corresponding element ist being sent (applies only to GSSE control blocks)</p>
SMVSettings	<p>The SMV control block attributes for which online setting is possible with service SetMSVCBValues respective SetUSVCBValues:</p> <p>The attribute's meaning is:</p> <p>cbName – control block name</p> <p>datSet – data set reference</p> <p>svID – sample value identifier</p> <p>optFields – optional fields to include in sample value message</p> <p>smpRate – sample rate</p>
ConfLNns	<p>Describes what can be configured for LNs defined in an ICD file</p> <p>The attribute meanings are:</p> <p>fixPrefix – if false, prefixes can be set/changed</p> <p>fixLnInst – if false, LN instance numbers can be changed</p>
ConfLdName	<p>If this element is present, the IED allows as a server to define functional LD names (by means of the LDevice ldName attribute), and a client can receive and interpret correctly SCL / online messages using this capability.</p>
GSEDir	<p>GSE directory services according to IEC 61850-7-2. This capability has no attributes.</p>
GOOSE	<p>This element shows that the IED can be a GOOSE server or client according to IEC 61850-7-2.</p> <p>The attributes meaning is:</p> <p>max = maximum number of GOOSE control blocks, which are configurable for publishing (max=0 means the device is only a GOOSE client).</p>
GSSE	<p>This element shows that the IED can be a binary data GSSE server or client according to IEC 61850-7-2.</p> <p>The attributes meaning is:</p> <p>max – maximum number of GSSE control blocks, which are configurable. Max=0 means only GSSE client.</p>
SMV	<p>This element shows that the IED can be a Sampled Value server or client according to IEC 61850-7-2.</p> <p>The attributes meaning is:</p> <p>max = maximum number of SMV control blocks, which are configurable for publishing (max=0 means the device is only a SMV client).</p>
FileHandling	<p>All file handling services; without attributes.</p>
<p>NOTE Within an IED capability description, the maximum numbers specified above shall be a guaranteed (minimal) 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).</p>	

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

<xs:complexType name="tAccessPoint">

```

<xs:complexContent>
  <xs:extension base="tNaming">
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element name="Server" type="scl:tServer">
          <xs:unique name="uniqueAssociationInServer">
            <xs:selector xpath="/scl:Association"/>
            <xs:field xpath="@associationID"/>
          </xs:unique>
        </xs:element>
        <xs:element ref="scl:LN" maxOccurs="unbounded"/>
        <xs:element name="ServerAt" type="scl:tServerAt"/>
      </xs:choice>
      <xs:element name="Services" type="tServices" minOccurs="0" />
      <xs:element name="GOOSESecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
      <xs:element name="SMVSecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
    </xs:sequence>
    <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 described by one of the elements: *Server*, *ServerAt* or *LN* list. It may optionally contain the security related elements *GOOSESecurity* and *SMVSecurity*, stating the certificate information to be used for GOOSE sending and Sampled Value sending. The detailed meaning of these certificate description can be found in IEC 62351-6.

The attributes of the Access point element are defined in Table 12.

**Table 12 – 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.

The *Services* element specifies the service capabilities of this access point additionally to those already stated at the IED level. It shall not contain IED general configuration or engineering related capabilities – these shall be defined at IED level. This might mean e.g. that the *ReportSettings* capability can appear with values "Conf" at the IED level, and with values "Dyn" at an access point, if another access point without this reporting capability exists at the same IED. Further the *max* attribute of the *GOOSE* and *GSSE* elements refers to the maximum number of GOOSE control blocks allowed to send via this access point, while the appropriate elements at IED level specify the maximum number of GOOSE control blocks configurable at the whole IED.

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

The *ServerAt* element references an existing access point, which shall contain a server. It can be used to define another access point to the same server. It has to be taken in mind, that this other access point shall be connected to other Subnetworks than all other access points of this server, and that all access points share all control block instances of the defined server. This means especially, that if a GOOSE message shall have different addresses at different Subnetworks, then another GOOSE control block instance shall be used.

```
<xs:complexType name="tServerAt">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:attribute name="apName" type="tName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *ServerAt* element has only the attribute *apName*, which references the *AccessPoint* on the same IED, which hosts / defines the server's data model.

The LN list allows to define a list of client LNs for a pure client access point. These client LNs can then be referenced to define the data flow between IEDs and to reserve control block instances, e.g. as reporting client to a report control block instance.

### Restrictions

- The name of the access point shall be unique within the IED.
- The name shall not be empty.
- The *GOOSESecurity* element and the *SMVSecurity* element are only allowed, if the attribute *certificate* of the server's *Authentication* element is *true*.
- The *ServerAt* element references a server access point on the same IED, which then shall contain the server's data model.

Note that

- an IED can be purely a router, switch 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, the switch function is not visible at the 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 is 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 13.

**Table 13 – 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 all attributes are missing, the default method is *none* (i.e. 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 14.

**Table 14 – Attributes of the Authentication element**

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

Note: The *GOOSESecurity* and *SMVSecurity* elements of the access point are only allowed to be used, if *certificate="true"* at the *Authentication* element.

### 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 name="ldName" type="tRestrLdName" use="optional"/>
    </xs:extension>
  </xs:complexContent>
```

```
</xs:complexType>
```

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

**Table 15 – Attributes of the LDevice element**

Attribute name	Description
inst	Identification of the LDevice within the IED. Its value cannot be the empty string. It is always used as key part for references to logical devices within the SCL file.
desc	The description text
ldName	The explicitly specified name of the logical device according to IEC 61850-7-1 and 7-2 within the communication. If missing, the default is the IED name concatenated with the inst value defined above.

### Restrictions

- The LD *inst* shall be unique within the IED.
- The LD name built from *inst* and other parts as described in 8.5 shall be unique within each SCL file.
- The *ldName*, if specified, must be unique within each SCL file, and different to any default name of other LDs.
- The length of the attribute *inst* shall be at least one character.
- The length of the logical device name (either *ldName*, or IED name concatenated with *inst*) is restricted to 64 characters, and it is alphanumeric with only underscore (\_) as additional character,

### 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: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), and *SCLControl*. Furthermore, it inherits *ReportControl*, *Log* and *LogControl* from the base type *tAnyLN*, as well as the *DOI* and *Inputs* element.

The *Log* element indicates that the logical device, which is controlled by this LN0, contains a log, and its name can be used as log name within a log control block.

The *SCLControl* element indicates that the logical device, which is controlled by this LN0, contains an SCL control block as defined in IEC 61850-8-1.

The attributes of the LN0 element are defined in Table 16.

**Table 16 – Attributes of the LN0 element**

Attribute name	Description
InClass	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
InType	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. For the referencing of links to LN0, *lnInst* shall be missing, and *InClass* 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="InClass" type="tLNClassEnum" use="required"/>
      <xs:attribute name="inst" type="tLNInst" use="required"/>
      <xs:attribute name="prefix" type="tPrefix" 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:unique name="uniqueSDI_DAIinDOI">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="Inputs" type="tInputs" minOccurs="0">
          <xs:unique name="uniqueExtRefInInputs">
            <xs:selector xpath="./scl:ExtRef"/>
            <xs:field xpath="@iedName"/>
            <xs:field xpath="@ldInst"/>
            <xs:field xpath="@prefix"/>
            <xs:field xpath="@InClass"/>
            <xs:field xpath="@lnInst"/>
            <xs:field xpath="@doName"/>
            <xs:field xpath="@daName"/>
            <xs:field xpath="@intAddr"/>
          </xs:unique>
        </xs:element>
        <xs:element name="Log" type="scl:tLog" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="InType" 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). Further it contains Log element(s), if the LN contains one or more logs. The log has as only attribute its name, which shall be unique within the LN.

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

**Table 17 – Attributes of the LN element**

Attribute name	Description
desc	The description text for the logical node
InType	The instantiable type definition of this logical node, reference to a LNodeType definition
InClass	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*, *InClass* 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.
- The *inst* attribute shall be a number with no more than 7 digits.
- The *prefix* follows the restrictions stated in IEC 61850-7-2.
- The *Log* element is only allowed in LLN0, and in special LN classes explicitly defined in other parts of this standard or in other standards.

### 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:unique name="uniqueSDI_DAIinSDI">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="DAI" type="tDAI"/>
      </xs:choice>
      <xs:attribute name="name" type="tDataName" 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 described by one of the element: *SDI* or *DAI*.

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

**Table 18 – 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; shall not be used if DOI has no 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 additional (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="tAttributeNameEnum" 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>

```

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 19.

**Table 19 – 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="tAttributeNameEnum" 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 20.

**Table 20 – 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 for DAI and SDI

- The name shall begin with a lower-case letter, except SIUnit and the exceptions defined in IEC 61850-8-1.
- Dots (.) are not allowed within names, only alphanumeric characters

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:choice maxOccurs="unbounded">
        <xs:element name="FCDA" type="tFCDA"/>
        <xs:element name="FCCB" type="tFCCB"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The DataSet contains a sequence of FCDA or FCCB elements. FCDA elements which reference attributes within a control block are called FCCB. The data set definition of the LN has the following attributes:

**Table 21 – 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="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="optional"/>
  <xs:attribute name="lnInst" type="tLNInst" 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:attribute name="ix" type="xs:unsignedInt" use="optional"/>
</xs:complexType>
```

The FCDA element defines the name of a functionally constrained data or functionally constrained data attribute according to IEC 61850-7-2 of this IED to be contained in the data set. The element has the following attributes:

**Table 22 – Attributes of the FCDA element**

Attribute name	Description
ldInst	The LD where the DO resides; shall always be specified except for GSSE
prefix	Prefix identifying together with <i>lnInst</i> and <i>lnClass</i> the LN where the DO resides; optional, default value is the empty string
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 <i>doName</i> is empty, then <i>fc</i> 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 missing, all attributes with functional characteristic given by <i>fc</i> 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 <i>fc</i> definition in 9.5
ix	An index to indicate an array element in case that one of the data elements is an array

```
<xs:complexType name="tFCCB">
  <xs:attribute name="ldInst" type="tName" use="required"/>
  <xs:attribute name="prefix" type="tAnyName" use="optional"/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
  <xs:attribute name="lnInst" type="tName" use="optional"/>
  <xs:attribute name="cbName" type="tName" use="required"/>
  <xs:attribute name="daName" type="tName" use="optional"/>
  <xs:attribute name="fc" type="tServiceFCEnum" use="required"/>
  <xs:attribute name="mustUnderstand" type="xs:boolean" use="required" fixed="true" />
</xs:complexType>
```

The FCCB element defines the name of a functionally constrained data or functionally constrained data attribute according to IEC 61850-7-2 of this IED to be contained in the data set and referencing a control block. The element has the following attributes:

**Table 23 – Attributes of the FCCB element**

Attribute name	Description
ldInst	The LD where the DO resides

prefix	Prefix identifying together with <i>InInst</i> and <i>InClass</i> the LN where the DO resides
InClass	LN class of the LN where the DO resides; shall always be specified.
InInst	Instance number of the LN where the DO resides; shall be specified except for LLN0
cbName	A name identifying the control block (within the LN).
daName	The attribute name – if missing, all attributes with functional characteristic given by <i>fc</i> 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. For the FCCB element only control block related <i>fc</i> 's are allowed.
mustUnderstand	Value is true; if this element appears within a data set definition, any tool / client processing this definition must understand this element.

Note: for backward compatibility the FCCB element should not be used in preconfigured data sets on an IED's ICD file. The *mustUnderstand* attribute shows that it must be understood to be able to receive data based on this data set.

Observe that a data set with an FCCB shall not be used by the control block referenced by the FCCB, else a loop back event generation flooding the IED (and the communication system) can be generated.

The order of data within a message based on this data set definition shall be the FCDA / FCCB order in the data set. If an FCDA specifies a set of attributes, for example via *fc*, then the order of data values is specified by the DATA definition and the attribute order in the corresponding LNs *LNNodeType*.

### Restrictions

- 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 *LNNodeType* 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.
- FCCB definitions are not allowed in data sets for GSSE control blocks.
- 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).
- A data set containing an FCCB shall not be used by the control block referenced by the FCCB, else a loop back event generation flooding the IED (and the communication system) can be generated.

### Data set example

```
<DataSet name="Example">
  <FCCB IdInst="C1" InClass="LLN0" fc="RP" cbName="PosReport"/>
  <FCDA IdInst="C1" prefix="" InInst="1" InClass="CSWI" doName="Pos" fc="ST"/>
  <FCDA IdInst="C1" prefix="" InInst="2" InClass="CSWI" doName="Pos" fc="ST"/>
  <FCDA IdInst="C1" prefix="" InInst="1" InClass="MMXU" doName="A" fc="MX"/>
  <FCDA IdInst="C1" prefix="" InInst="1" InClass="MMXU" doName="PhV" fc="MX"/>
</DataSet>
```

### 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="optional"/>
    <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:attribute name="indexed" 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 report control block (RCB) contains the elements: *TrgOps*, *OptFields* and *RptEnabled*.

The attributes given in Table 24 are used.

**Table 24 – 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 should only be missing within an ICD-File, or to indicate an unused control block.
intgPd	Integrity period in milliseconds – see IEC 61850-7-2. Only relevant if trigger option <i>period</i> is set to true
rptID	Identifier for the report control block, optional; if not used, or value is the empty string (only for backward compatibility), then the IED has to take the default value as defined in IEC 61850-7-2
confRev	The configuration revision number of this report control block
buffered	Specifies if reports are buffered or not – see IEC 61850-7-2; default: false
bufTime	Buffer time – see IEC 61850-7-2; default: 0
Indexed	If true, the report control block instance names are built from the supplied name, followed by an index number from 01 up to maximum 99

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:attribute name="gi" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="includeAll" 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 only exception is the *gi* trigger option, which per default is true due to backwards compatibility reasons.

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="entryID" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="configRef" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="bufOvfl" type="xs:boolean" use="optional" default="true"/>
</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 attribute `bufOvfl` is per default true, i.e. it has only to be set if it shall be false. The attribute `segmentation` is deprecated, because it has no meaning, and is removed from this version. It shall however be accepted as input for backward compatibility and should not be used respective be ignored at input.

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 25 are used.

**Table 25 – 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). The default value is 1. A missing <code>RptEnabled</code> element within an ICD file indicates that this value shall be set by the system configuration tool within the limits defined by the <code>ConfReportControl</code> and <code>DynAssociation</code> element's <code>max</code> attributes.

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 is instantiated in the IED. Observe that for all permanently used buffered control blocks, a `ClientLN` shall be preconfigured, and `ResvTms` set to -1, if it exists. For all other control block instances an existing `ResvTms` shall be set to 0. 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 and the attribute `indexed` is set to `true` (which is the default value), 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 01). 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 for 99 clients. In case of unbuffered control blocks *indexed* may be set to *false*, meaning that each client gets a 'virtual' RCB instance with the name as specified by the *name* attribute.

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="InClass" type="tLNClassEnum" use="required"/>
  <xs:attribute name="InInst" type="xs:normalizedString" use="required"/>
</xs:attributeGroup>
```

The attributes given in Table 26 are used.

**Table 26 – Attributes of the ClientLN element**

Attribute name	Description
iedName	The name of the IED where the LN resides
IdInst	The instance identification of the LD where the LN resides
prefix	The LN prefix
InClass	The LN class according to IEC 61850-7-4
InInst	The instance id of this LN instance of below LN class in the IED
desc	optional descriptive text, e.g. about purpose of the client

### ClientLN restrictions

- Both, the iedName and IdInst (as specified in the attribute group agLDRef) shall be specified with non zero length. If the reference is to an LN at a pure client access point, then the value of IdInst can be arbitrary (recommended: LD0).
- Only the prefix is optional, in case that the referenced LN instance has no prefix (prefix value = empty string). If it is defined, it must have a non zero length.
- InInst is required. If the LLN0 is referenced, the value is the empty string.

### Report control block restrictions

- The name of the report control block shall be unique within the LN. It contains only alphanumeric characters.
- The datSet attribute must contain a valid reference. If an unused control block shall be in the IED, then for this the datSet attribute must be left out completely.
- The rptID can be missing, if the default value according to IEC 61850-7-2 shall be used. However, if the attribute is used, its value should not be the empty string (however the empty string shall be accepted as input for backward compatibility).

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 a separately supplied IdName. 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 as defined in 9.3.14.

Example:

```
<ReportControl name="PosReport" rptID="E1Q1Switches" datSet="Positions" confRev="0">
  <TrgOps dchg="true" qchg="true"/>
  <OptFields/>
  <RptEnabled max="5">
    <ClientLN iedName="A1KA1" ldInst="LD0" lnInst="1" lnClass="IHMI"/>
  </RptEnabled>
</ReportControl>
```

The RptEnabled part defines that the Report control block type is valid for 5 (unbuffered; missing *buffered* attribute) RCBs with names PosReport01, PosReport02, up to PosReport05 (missing *indexed* attribute means *true*). 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="ldInst" type="tLDInst" use="optional"/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
      <xs:attribute name="lnClass" type="scl:LNClassEnum" use="optional" default="LLN0"/>
      <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
      <xs:attribute name="logName" type="tLogName" 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>
```

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 27.

**Table 27 – 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 should only be missing within an ICD-File, or for an unused control block.
intgPd	integrity scan period in milliseconds – see IEC 61850-7-2.
ldInst	The identification of the LD where the log resides; if missing, the same LD where this control block is placed.
prefix	Prefix of LN where the log resides; if missing, empty string
lnClass	Class of the LN where the log resides; if missing, LLN0
lnInst	Instance number of LN, where the log resides; missing for LLN0
logName	Relative name of the log within its hosting LN; name of the log element
logEna	TRUE enables immediate logging; FALSE prohibits logging until enabled online
reasonCode	If true, the reason code for the event trigger is also stored into the log – see IEC 61850-7-2

## Restrictions

- The name of the log control block shall be unique within the LN.
- The datSet attribute shall contain a valid data set reference, or be missing completely.
- The log reference shall point to a valid, defined log

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

```
<LogControl name="LogPos" 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 28 are used.

**Table 28 – 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 should only be missing within an ICD-File, or to indicate an unused control block.
confRev	The configuration revision number of this control block. It is recommended to increment this by 10 000 on each configuration change, to distinguish this from online changes leading to an increment of 1 only.
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.
- The datSet attribute must contain a valid data set reference, or be missing completely.
- The confRev attribute is mandatory if the type is GOOSE (respective the type attribute is not specified).
- Different applications within the station shall have unique appld 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" applID="Itl" " confRev="20000" />
```

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 29 are used.

**Table 29 – 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 should only be missing within an ICD-File, or to indicate an unused control block
confRev	The configuration revision number of this control block; mandatory. It is recommended to increment it by 10000 on any configuration change, to distinguish this from online configuration changes leading to an increment of 1 only.
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	<i>false</i> 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 attribute `datSet` must contain a valid data set reference, or be missing completely,
- The `UsvCBName` defined in IEC 61850-7-2 shall be set to this IED's name concatenated with the `name` above (which might then be empty),
- the `Resv` attribute of the CB as defined in IEC 61850-7-2 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="sampleRate" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="dataSet" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="security" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
```

The attributes of the Smv Options element are defined in Table 30.

**Table 30 – 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
sampleRate	
dataSet	The meaning of the options is described in IEC 61850-7-2. If the attribute is set to true, the dataset name shall be included into the SMV telegram
security	See IEC 61850-9-2 for description

The following options are deprecated and removed from the 2007 version, however should be accepted as input for backward compatibility.

**Table 31 - Deprecated Smv options:**

Attribute name	Description
dataRef	This value is no longer supported in SV-Telegrams. Therefore only the value false can be accepted.
sampleSynchronized	This value is now always in the SV telegrams. The option is kept in the syntax for backward compatibility. Only value true can be accepted.

## Restrictions

- The SV control block name shall be unique within the LLN0, i.e. within the LDevice.
- The `confRev` attribute is mandatory for the SV control block

The following SCL extract show the definition of a 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" />
</SampledValueControl>
```

### 9.3.12 Setting control block

The following defines the definition for a setting group control block (SGCB). Note that the SGCB name, i.e. its name part within the LN0, is SGCB according to IEC 61850-7-2. Therefore, only one SGCB 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 32.

**Table 32 – 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, i.e. signals sent from other LNs possibly on other IEDs, which are needed by the LN application to fulfill its function. The section also allows the binding of the signal to an IED internal address IntAddr.

```

<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 IntAddr 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="scl:agDesc"/>
  <xs:attribute name="iedName" type="tIEDName" use="optional"/>
  <xs:attribute name="ldInst" type="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="scl:tPrefix" use="optional"/>
  <xs:attribute name="lnClass" type="scl:LNClassEnum" use="optional"/>
  <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
  <xs:attribute name="doName" type="scl:tName" use="optional"/>
  <xs:attribute name="daName" type="scl:tName" use="optional"/>
  <xs:attribute name="intAddr" type="xs:normalizedString" use="optional"/>
</xs:complexType>

```

The attributes shown in Table 33 are used.

**Table 33 – 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
InClass	The LN class according to IEC 61850-7-x
InInst	The instance id of this LN instance of above LN class in the IED; missing for a reference in LLN0. For backwards compatibility also the empty string shall be accepted for LLN0.
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 (IntAddr) for all process input attributes of a DO ( <i>fc</i> = 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
desc	A free description / text. Can e.g. be used at system engineering time to tell the IED engineer the purpose of this incoming data.

If all attributes except *intAddr* (and possibly *desc*) are missing, this allows to specify the available internal addresses to bind external references in advance. The *desc* attribute allows to give it some meaning. If any of the other attributes is specified, then a missing *daName* attribute means all the operational value attribute(s) of the DO, i.e. *stVal*, *mag*, etc. In this case, *intAddr* can also specify the addresses of all operational attributes in some IED tool specific way. So, both usages are supported additionally to a full link specification: only *intAddr* e.g. within an ICD file, or only an external reference by means of *iedName/IdInst/prefix/InClass/InInst* coming from an SCD file. The *desc* attribute can be used to indicate the purpose of this link, dependent on the usage above.

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.

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 34 are used.

**Table 34 – 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
IdInst	The reference to the client logical device
InClass	The class of the client LN
prefix	The LN prefix
InInst	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 22 gives an overview of the Communication section.

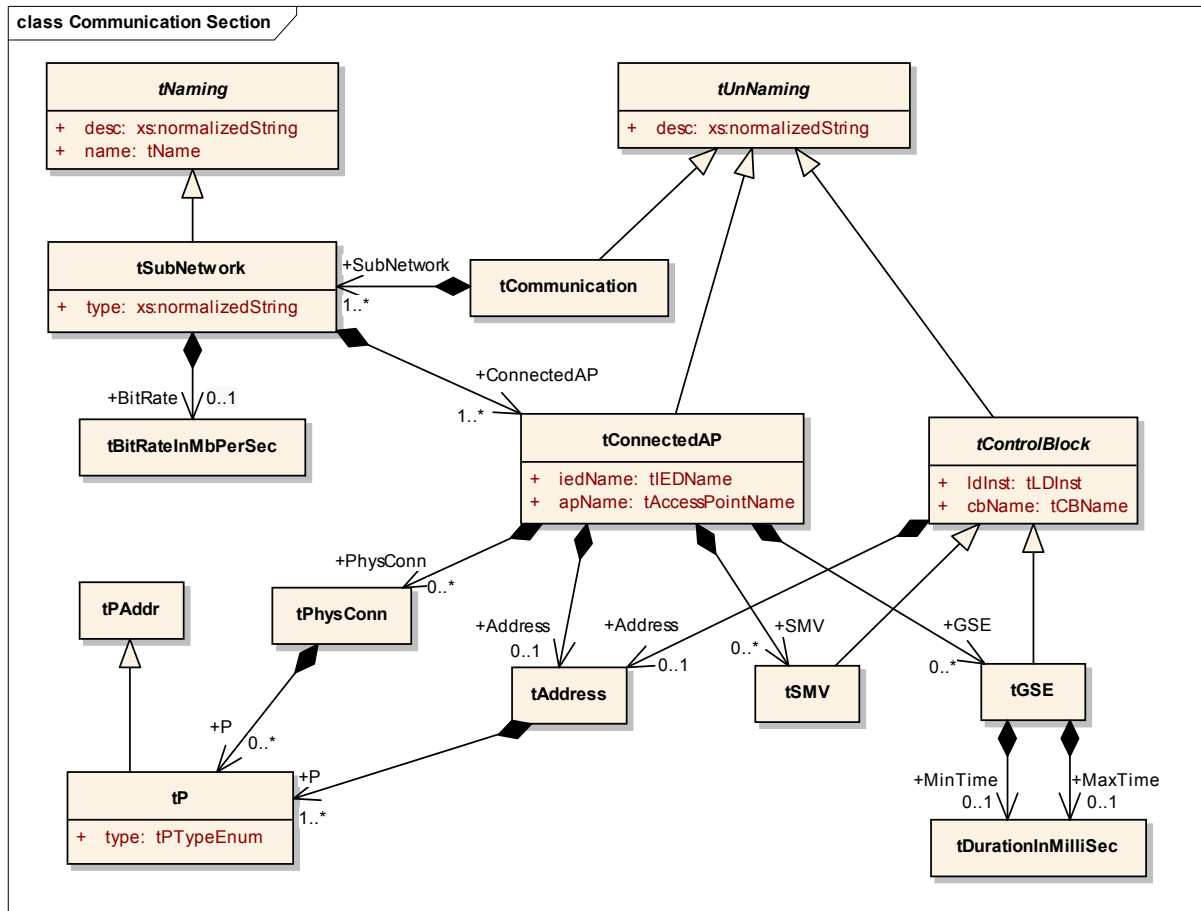


Figure 22 – UML diagram overview of the Communication section

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 35.

**Table 35 – 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; IP should be used for all IP based protocols except those explicitly standardized.

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. Additionally the type IP is predefined for all IP based protocols except those specifically standardized, to allow unique IP address checking across all protocols (subnetworks) on the same (physical) network.

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 36.

**Table 36 – 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 *IdInst* 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="IdInst" type="tLDInst" use="required"/>
      <xs:attribute name="cbName" type="tCBName" 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 37.

**Table 37 – Attributes of the GSE element**

Attribute	Description
desc	Textual description
IdInst	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 IdInst.

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 38.

**Table 38 – Attributes of the SMV element**

Attribute	Description
desc	Textual description.
IdInst	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 IdInst.

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"). The *PhysConn* type *Connection* defines a first physical connection, while the type *RedConn* can identify additional physically redundant connection(s). 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 39 shall be used.

**Table 39 – PhysConn P-Type definitions**

PhysConn type	P type	Recommended values (IEC 61850-8-1 related)
Connection, RedConn	Type	<i>10BaseT</i> , <i>100BaseT</i> etc. for electrical connection <i>FOC</i> for optical connection <i>Radio</i> for radio connection, for example WLAN
	Plug	<i>RJ45</i> for electrical plug <i>ST</i> for bajonet plug (optical glass)
	Cable	The identification of a physical cable for this connection, which connects this access point to another access point
	Port	The identification of a port or terminal at this access point to which a cable is connected (see above) or may be connected

## Restrictions

- The PhysConn *type* values as well as its P parameter *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 W01, 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 IdInst="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 IdInst="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 respective DOType's 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*. 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 generate a new LN type name by concatenating the IED name (which shall be unique within the file) with the old LNodeType name (which shall be unique at least per IED). If a LN type is generally valid for several IEDs of different type, then the *iedType* attribute shall be defined as empty string. If it is important to keep the relation of the LNodeType to the IED type, then *iedType* should be set to the same value as the IED's type attribute. Especially if an IED configurator needs the LNodeType contents back unchanged, it shall bind the LNodeType to the IED type by setting the *iedType* attribute identical to the IED's type attribute.

The order of DO elements within a LNodeType definition, and of SDO/DA elements (see 9.5.3) 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.

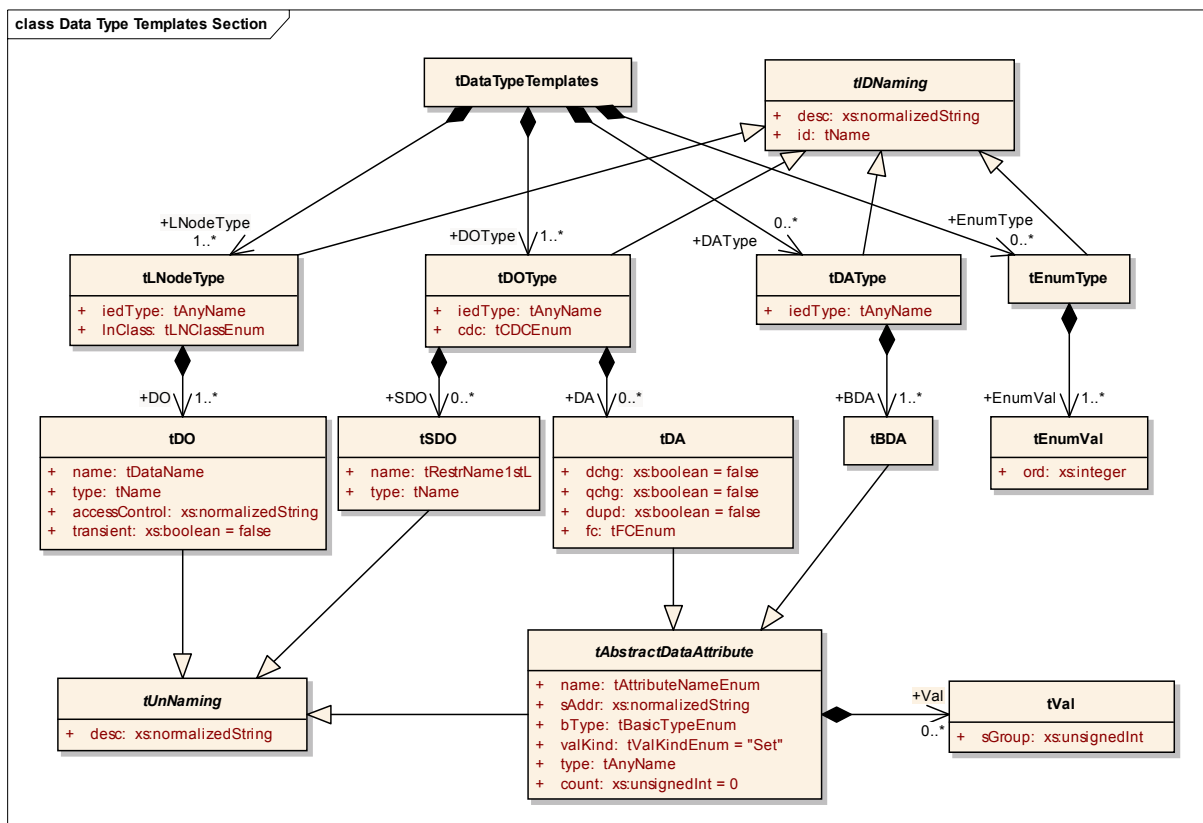


Figure 23 – UML overview of DataTypeTemplate section

Here follows the XML schema definition inclusive defined restrictions within *Data Type Templates*.

```

<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 40 can appear there.

**Table 40 – 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 <i>bType</i> 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="InClass" type="tLNClassEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The attributes have the following meaning as shown in Table 41.

**Table 41 – Attributes of the LNodeType element**

Attribute	Description
id	A reference identifying this LN type within this SCL section; used by the LN attribute LNTType 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
InClass	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 42.

**Table 42 – Attributes of the DO element**

Attribute	Description
name	The DATA name as specified for example in IEC 61850-7-4.
type	The <i>type</i> references the <i>id</i> 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 43.

**Table 43 – Attributes of the DOType element**

Attribute	Description
id	The (global) identification of this DOType within an <i>iedType</i> . 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 without IED identification.
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="tRestrName1stL" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tDA">
```

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

**Table 44 – 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 attribute 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 45.

**Table 45 – 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	<i>false</i> , <i>true</i> 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
ObjRef	normalizedString	The reference to an IEC 61850 object, as defined in IEC 61850-7-2
Timestamp (UTC time)	dateTime	Coding without time zone, e.g. 2007-12-31T21:01:12.345

NOTE It is not intended to specify values of types 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 46 (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:restriction>
</xs:simpleType>
```

```

    <xs:enumeration value="Set"/>
  </xs:restriction>
</xs:simpleType>

```

**Table 46 – 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 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, e.g. for startup or simulation
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 tAbstractDataAttribute 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 47.

**Table 47 – 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=SE always also implies fc=SG; fc=SG means that the values are visible, but not editable; if an attribute has ST and CO respectively MX and CO in IEC 61850-7-3, 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 ctModel 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 46.

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="VisString129"/>
    <xs:enumeration value="VisString255"/>
    <xs:enumeration value="Unicode255"/>
    <xs:enumeration value="Octet64"/>
    <xs:enumeration value="Struct"/>
    <xs:enumeration value="EntryTime"/>
    <xs:enumeration value="Check"/>
    <xs:enumeration value="ObjRef"/>
  </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>

```

*tPredefinedBasicTypeEnum* contains the definitions as defined in IEC 61850-7-x. CODED ENUMs are replaced 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. Check is introduced for the appropriate data attribute used in IEC 61850-8-1. While Quality and Check remain opaque (no values required in SCL), Dbpos and Tcmd are handled like Enums for the value coding, and the identification of the enumeration is identical to the bType. For VisibleString, UnicodeString and OctetString length dependent (sub-)types are introduced. VisString32 is for example a VisibleString of maximum length of 32 characters. *ObjRef* is basically a string type, which contains the reference to another IEC 61850 object as defined in IEC 61850-7-2, where also the maximum allowed length is specified.

*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. Note that to check basic type conformance against IEC 61850-7-3 already by using an XML schema, *tPredefinedBasicTypeEnum* can be used instead of *tBasicTypeEnum* for the *bType* attribute in the syntax type *tAbstractDataAttribute* in 9.5.4.1.

*tPredefinedBasicTypeEnum* will be used for the schema of this version. It must be kept in mind that, e.g. after extensions in other standards, as input also other types according to *tBasicTypeEnum* should be syntactically accepted – and must be handled e.g. with *mayIgnore* or *mustUnderstand* rules.

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. The XML coding of the value is defined in 9.5.4.1 respective Table 45. 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 / LD, 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: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: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 48.

**Table 48 – 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 <i>bType</i> = Enum or <i>bType</i> = 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 46.

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, INS or ING common data classes. Therefore both, the status value and 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 0. 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 49.

**Table 49 – Attributes of the EnumType element**

Attribute	Description
id	A reference identifying this enumeration type; used by the <i>type</i> attribute of DA and BDA elements to reference this definition in the case where the <i>bType</i> 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:attributeGroup ref="agDesc"/>
    </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. The *desc* attribute allows descriptive text for the meaning of the value.

### 9.5.7 Data type template examples

Examples can be found in the *Data Type Template* section of Clause D.2.

## 10 Tool and project engineering rights

This clause refers to the general clause 5 about the engineering process, which defines the roles of a system configurator and an IED configurator within a system project, and additionally a communication interface related data exchange at system level between different projects. This clause defines the intended engineering responsibility areas in terms of the previously defined SCL elements.

### 10.1 IED configurator

The task of the IED configurator is to create the ICD file, and to modify the data model, parameter and configuration values either for a new ICD file, or a project specific IED instance by means of an IID file. Both may contain preconfigured data sets and control blocks, and default addresses for an IED of this type. For an IID file produced from an SCD file, the data sets and control blocks shall remain unchanged against the SCD file. Finally the IED configurator is responsible to bind incoming data from other IEDs as defined within an imported SCD file to internal signals, e.g. by means of the SCL Input section, and generate and load the IED instance specific configuration data, where a CID file could be a part of.

How an ICD file is created, depends on the IED capabilities and the tool design. There is a big range from a fix ICD file to be used in each project (the only possible adaptations are the IED name respective LD name and IED address), up to ICD file generation for a specific usage of the IED e.g. for a project specific bay type after extensive preengineering by means of the IED tool – see also clause 7 – or a preengineered project specific IED based on the current state of the system (IID file generated based on a SCD file).

Any change in data model, parameter and configuration values shall be reflected in appropriate version indications within the LN0 NamPit DATA as values within the SCL file. For predefined data sets and control blocks the version information has to be managed as defined in IEC 61850-7-2 for the *confRev* parameter of the control blocks, and also contained in the SCL file.

If an SCD file is imported, an IED tool may update version and related value information and change parameter and configuration values as well as the binding of external data to internal

signals. It may add new control blocks and data sets. The result is transferred back to the system configurator by means of an IID file. Only the following data model changes are allowed (but not mandatory):

- Addition of logical devices, logical nodes or DATA within logical nodes
- Removal of logical devices, logical nodes or DATA, which are not referenced by some client or bound to the primary system description (substation section)

Observe that both kinds of changes lead to new data model version identifications, which have to be reimported by the system tool and might influence the reloading of other IEDs in the system.

Note: IED online change of data flow is described and defined in IEC61850-7-2. Online changes of data models on an IED outside SCL are out of scope of the standard. To keep system consistency for IEDs already integrated into a system, they should either be prevented or follow the above rules, especially providing new data model revision information.

## 10.2 System configurator

The tasks of the system configuration tool is to create IED instances from IED templates, engineer the data flow between the IEDs, give addresses to them, and bind the logical nodes to the primary system. Therefore beneath instantiating IED templates, the system tool handles the following SCL sections:

- Substation section, including references to logical nodes on IEDs (9.2)
- Communication section inclusive project specific instance addresses (9.4)
- Data sets and control blocks, as allowed by the IED capabilities (9.3.7 and following)
- Allocation of data flow and report control block instances to clients, as allowed by the IED capabilities (ClientLN element at report control blocks, IEDName at other control blocks).
- Creating IED input sections as seen from system engineering point of view, however without binding to IED internal signals (9.3.13).
- Reorganizing the DataTypeTemplate section (9.5) to keep the type identifiers unique and the template section short, however under the constraint, that the instance information is unchanged when the templates are expanded at IEDs to an instance. This concerns not only the structures and type, but also the values and Private sections defined within the types.

The system configuration tool shall increment the control block *confRev* values of all statically defined data sets on creation and modification as defined in IEC 61850-7-2. Further it defines the project identity by means of the SCD Header identification, and manages the SCD header revision history.

In case that a system configurator also changes configuration values and parameter values for an IED, it has also to increment the appropriate *paramRev* and *valueRev* attributes in the LN0 *NamPit*. It is the responsibility of the system engineer to clarify before such changes, whether the concerned IED supports loading of this data via an SCD file.

## 10.3 Right transfer between projects

The right of data flow engineering can be formally transferred from one project to another project by means of a SED file. The concerned IEDs are marked with the *engRight* attribute

value *dataflow*. To not lose already predefined references on these IEDs, all referenced IEDs have also to be exported at least partly, i.e. just the LDs and data sets referenced, with *engRight=fix*. If address coordination is an issue, also the access points of all IEDs with defined addresses can be exported.

Observe that also the appropriate part of the Substation section and Communication section shall be exported, however only as far as they contain references to the exported IEDs, or shall be used by the other project. The primary part shall contain topologically and equipment wise complete bays. The relation between exported IED's logical nodes and the primary equipment is fix for the receiving project, however it might add references to its own IEDs into the Substation section, and shall add addresses of its own IEDs into the communication section.

The importing project can use all received information as needed (e.g. use one of the fix IEDs as client for its own IEDs), however has only the following engineering rights (modification rights) on those IEDs exported with *dataflow* right:

- Addition of data sets and control blocks as allowed by the IED capabilities. Observe that the IED owner is allowed to restrict these capabilities further.
- Allocation of additional clients to control blocks
- Addition of data expected to be received from its own IEDs to the Input sections

When ready with engineering, the receiving project has to export again a SED file, containing all imported IEDs plus those of its own project referenced by them after engineering either as fix, or again with *dataflow* right. The exported SED file shall have the same Header identification as the imported SED file, however with increased revision index. This gives the originally exporting project the full engineering right back. Observe that if some IED with *dataflow* right of the other project is imported, that then again after engineering finalization a SED has to be exported back. It is up to the exporting project to decide, if its own IEDs are again exported with *dataflow* engineering right to have another engineering round, or just as fix to complete the data exchange.

Observe that IEDs exported as fix are still under full control of the exporting project and might be changed by it. If this happens, this is latest discovered online by discrepancies between assumed and actual control block and data model revision information. However, it is good engineering practice to notify the concerned project, if this is noticed, e.g. by sending another SED file.

IEDs exported as *dataflow* have to be set fix in their own project, which should block any changes. It is good practice to export this state also with SCD files and other SED files to other projects. This prohibits that two different projects add data flow definitions to the same IED at the same time.

## 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 xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.iec.ch/61850/2003/SCL"
targetNamespace="http://www.iec.ch/61850/2003/SCL" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema2007. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</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="tAcsiName">
    <xs:restriction base="xs:Name">
      <xs:pattern value="\p{L}[\d,\p{L},_]">
    </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{L}[\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:simpleType name="tScIVersion">
    <xs:restriction base="tName">
      <xs:pattern value="2\d{2}">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tScIRevision">
    <xs:restriction base="xs:Name">
      <xs:pattern value="\p{Lu}">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tRestrLdName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tEmpty">

```

```

        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="0"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tIEDName">
        <xs:restriction base="tAcsiName">
          <xs:maxLength value="64"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLDName">
        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="64"/>
          <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLDInst">
        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="64"/>
          <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLDInstOrEmpty">
        <xs:union memberTypes="tLDInst tEmpty"/>
      </xs:simpleType>
      <xs:simpleType name="tPrefix">
        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="6"/>
          <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
          <xs:pattern value=""/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLNInstOrEmpty">
        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="7"/>
          <xs:pattern value="[d]*"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLNInst">
        <xs:restriction base="tLNInstOrEmpty">
          <xs:minLength value="1"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tDataName">
        <xs:restriction base="tRestrName1stU">
          <xs:maxLength value="10"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tDataSetName">
        <xs:restriction base="tAcsiName">
          <xs:maxLength value="32"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tCBName">
        <xs:restriction base="tAcsiName">
          <xs:maxLength value="32"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tLogName">
        <xs:restriction base="tAcsiName">
          <xs:maxLength value="64"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="tAccessPointName">
        <xs:restriction base="xs:normalizedString">
          <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:schema>

```

File **SCL\_Enums.xsd**

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</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:enumeration value="Oper"/>
      <xs:enumeration value="SBO"/>
      <xs:enumeration value="SBOw"/>
      <xs:enumeration value="Cancel"/>
    </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:restriction>
  </xs:simpleType>

```

```

        <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:enumeration value="AB"/>
        <xs:enumeration value="BC"/>
        <xs:enumeration value="CA"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tAuthenticationEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="none"/>
        <xs:enumeration value="password"/>
        <xs:enumeration value="weak"/>
        <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="MHA1"/>
        <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:restriction>
</xs:simpleType>

```

```

    <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:restriction>
</xs:simpleType>

```

```

        <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="tServiceFCEnum">
    <xs:restriction base="xs:Name">
      <xs:enumeration value="SG"/>
      <xs:enumeration value="BR"/>
      <xs:enumeration value="RP"/>
      <xs:enumeration value="LG"/>
      <xs:enumeration value="GO"/>
      <xs:enumeration value="GS"/>
      <xs:enumeration value="MS"/>
      <xs:enumeration value="US"/>
    </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="VisString129"/>
      <xs:enumeration value="VisString255"/>
      <xs:enumeration value="Octet64"/>
      <xs:enumeration value="Unicode255"/>
      <xs:enumeration value="Struct"/>
      <xs:enumeration value="EntryTime"/>
      <xs:enumeration value="Check"/>
      <xs:enumeration value="ObjRef"/>
    </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:restriction base="tPredefinedBasicTypeEnum"/>
    <!--<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="m2"/>
        <xs:enumeration value="m3"/>
        <xs:enumeration value="m/s"/>
        <xs:enumeration value="m/s2"/>
        <xs:enumeration value="m3/s"/>
        <xs:enumeration value="m/m3"/>
        <xs:enumeration value="M"/>
        <xs:enumeration value="kg/m3"/>
        <xs:enumeration value="m2/s"/>
        <xs:enumeration value="W/m K"/>
        <xs:enumeration value="J/K"/>
        <xs:enumeration value="ppm"/>
        <xs:enumeration value="1/s"/>
        <xs:enumeration value="rad/s"/>
        <xs:enumeration value="VA"/>
        <xs:enumeration value="VAR"/>
        <xs:enumeration value="phi"/>
        <xs:enumeration value="cos_phi"/>
        <xs:enumeration value="Vs"/>
        <xs:enumeration value="V2"/>
        <xs:enumeration value="As"/>
        <xs:enumeration value="A2"/>
        <xs:enumeration value="VAh"/>
        <xs:enumeration value="Wh"/>
        <xs:enumeration value="VArh"/>
        <xs:enumeration value="V/Hz"/>
        <xs:enumeration value="Hz/s"/>
        <xs:enumeration value="char"/>
        <xs:enumeration value="char/s"/>
        <xs:enumeration value="kgm2"/>
        <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:simpleType name="tRightEnum">
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="full"/>
    <xs:enumeration value="fix"/>
    <xs:enumeration value="dataflow"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

File **SCL\_BaseTypes.xsd**

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema2007. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="SCL_Enums.xsd"/>
  <xs:attributeGroup name="agDesc">
    <xs:attribute name="desc" type="xs:normalizedString" use="optional" default=""/>
  </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="tAcsiNaming" abstract="true">
    <xs:complexContent>
      <xs:extension base="tBaseElement">
        <xs:attribute name="name" type="tAcsiName" use="required"/>
        <xs:attributeGroup ref="agDesc"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tAnyContentFromOtherNamespace" abstract="true" mixed="true">
    <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: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:complexContent mixed="true">
      <xs:extension base="tAnyContentFromOtherNamespace">
        <xs:attribute name="type" type="xs:normalizedString" use="required"/>
        <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" default=""/>
<xs:attribute name="toolID" type="xs:normalizedString"/>
<xs:attribute name="nameStructure" use="optional" default="IEDName">
  <xs:simpleType>
    <xs:restriction base="xs:Name">
      <xs:enumeration value="IEDName"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="tHitem" mixed="true">
  <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" default=""/>
    </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" default=""/>
    </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" default=""/>
    </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:restriction>
  </xs:simpleContent>
</xs:complexType>

```

```
<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 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/11/19.</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:unique name="uniqueLNodeInPowerTransformer">
              <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 name="GeneralEquipment" type="tGeneralEquipment" minOccurs="0"
maxOccurs="unbounded">
            <xs:unique name="uniqueLNodeInGeneralEquipment">
              <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: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:unique name="uniqueLNodeInSubEquipment">
      <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: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:unique name="uniqueLNodeInTransformerWinding">
            <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: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:unique name="uniqueLNodeInTapChanger">
            <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: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:unique name="uniqueLNodeInVoltageLevel">
            <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 name="Function" type="tFunction" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueSubFunctionInFunctionVL">
            <xs:selector xpath="/scl:SubFunction"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueGeneralEquipmentInFunctionVL">
            <xs:selector xpath="/scl:GeneralEquipment"/>
            <xs:field xpath="@name"/>
          </xs:unique>
          <xs:unique name="uniqueLNodeInFunctionSS">
            <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: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:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:unique name="uniqueChildNameInBay">
  <xs:selector xpath="/*"/>
  <xs:field xpath="@name"/>
</xs:unique>
<xs:unique name="uniqueLNodeInBay">
  <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 name="Function" type="scl:tFunction" minOccurs="0" maxOccurs="unbounded">
  <xs:unique name="uniqueSubFunctionInFunctionBay">
    <xs:selector xpath="/scl:SubFunction"/>
    <xs:field xpath="@name"/>
  </xs:unique>
  <xs:unique name="uniqueGeneralEquipmentInFunctionBay">
    <xs:selector xpath="/scl:GeneralEquipment"/>
    <xs:field xpath="@name"/>
  </xs:unique>
  <xs:unique name="uniqueLNodeInFunctionVL">
    <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: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:unique name="uniqueLNodeInConductingEquipment">
            <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 name="ConnectivityNode" type="tConnectivityNode" minOccurs="0"
maxOccurs="unbounded">
          <xs:unique name="uniqueLNodeInConnectivityNode">
            <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 name="Function" type="scl: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:unique name="uniqueLNodeInFunctionB">
            <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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:unique>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tLNode">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:attribute name="iedName" type="tIEDName" use="optional" default="None"/>
      <xs:attribute name="ldInst" type="tLDInstOrEmpty" use="optional" default=""/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
      <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
      <xs:attribute name="lnInst" type="tLNInstOrEmpty" use="optional" default=""/>
      <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:unique name="uniqueLNodeInSubFunction">
            <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 name="GeneralEquipment" type="tGeneralEquipment" minOccurs="0"
maxOccurs="unbounded">
          <xs:unique name="uniqueLNodeInGeneralEquipmentOfFunction">
            <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 name="ConductingEquipment" type="tConductingEquipment" minOccurs="0"
maxOccurs="unbounded">
          <xs:unique name="uniqueLNodeInConductingEquipmentOfFunction">
            <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:sequence>
      <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
    </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:unique name="uniqueLNodeInGeneralEquipmentOfSubFunction">
            <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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:element>
    <xs:element name="ConductingEquipment" type="scl:tConductingEquipment" minOccurs="0"
maxOccurs="unbounded">
      <xs:unique name="uniqueLNodeInConductingEquipmentOfSubFunction">
        <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:sequence>
  <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
</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" default=""/>
      <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:attribute name="neutralPoint" type="xs:boolean" use="optional" default="false"/>
    </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>
  <!-- This identity constraint must be removed, as there is a problem with the (according to part 6 text)
predefined connectivity node grounded resp GROUNDED. If a terminal references this node, which
is naturally NOT defined explicitly in the SCL file, verification fails. As far as I know, there is no way
to restrict the constraint in this respect, so that for value grounded it is allowed to have no explicit definition.
<xs:keyref name="ref2ConnectivityNode" refer="ConnectivityNodeKey">
  <xs:selector xpath="/scl:Terminal"/>
  <xs:field xpath="@connectivityNode"/>
</xs:keyref>
-->
  <xs:unique name="uniqueChildNameInSubstation">
    <xs:selector xpath="/*"/>
    <xs:field xpath="@name"/>
  </xs:unique>
  <xs:unique name="uniqueLNodeInSubstation">
    <xs:selector xpath="/scl:LNode"/>
    <xs:field xpath="@lnInst"/>
    <xs:field xpath="@lnClass"/>
    <xs:field xpath="@iedName"/>
    <xs:field xpath="@ldInst"/>
  </xs:unique>

```

```
<xs:field xpath="@prefix"/>  
</xs:unique>  
</xs:element>  
</xs:schema>
```

### A.3 Data type templates

#### File SCL\_DataTypeTemplates.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:scl="http://www.iec.ch/61850/2003/SCL" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.iec.ch/61850/2003/SCL" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</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="tNodeType">
    <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" default=""/>
        <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="tDataName" 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" default=""/>
        <xs:attribute name="cdc" type="tCDCEnum" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tSDO">
    <xs:complexContent>
      <xs:extension base="tUnNaming">
        <xs:attribute name="name" type="tRestrName1stL" use="required"/>
        <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" default=""/>
    </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:attributeGroup ref="agDesc"/>
    </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="uniqueDAorSDOInDOType">
        <xs:selector xpath="./"/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
    <xs:element name="DAType" type="tDAType" minOccurs="0" maxOccurs="unbounded">
      <xs:unique name="uniqueBDAInDAType">
        <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: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 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/11/19.</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" fixed="true"/>
    <xs:attribute name="sampleRate" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dataSet" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="security" 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="entryID" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="configRef" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="bufOvfl" type="xs:boolean" use="optional" default="true"/>
  </xs:attributeGroup>
  <xs:attributeGroup name="agLDRef">
    <xs:attributeGroup ref="scl:agDesc"/>
    <xs:attribute name="iedName" type="tIEDName" use="required"/>
    <xs:attribute name="ldInst" type="tLDInst" use="required"/>
  </xs:attributeGroup>
  <xs:attributeGroup name="agLNRef">
    <xs:attributeGroup ref="agLDRef"/>
    <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
    <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
    <xs:attribute name="lnInst" type="tLNInstOrEmpty" 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="tUnNaming">
        <xs:sequence>
          <xs:element name="Services" type="tServices" minOccurs="0"/>
          <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
            <xs:unique name="uniqueLNlnAccessPoint">
              <xs:selector xpath="/scl:LN"/>
              <xs:field xpath="@inst"/>
              <xs:field xpath="@lnClass"/>
              <xs:field xpath="@prefix"/>
            </xs:unique>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
    <xs:attribute name="name" type="tIEDName" use="required"/>

```

```

    <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:attribute name="originalSclVersion" type="tSclVersion" use="optional"/>
    <xs:attribute name="originalSclRevision" type="tSclRevision" use="optional"/>
    <xs:attribute name="engRight" type="tRightEnum" use="optional" default="full"/>
    <xs:attribute name="owner" type="xs:normalizedString" use="optional"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tServices">
  <xs:all>
    <xs:element name="DynAssociation" type="tServiceWithOptionalMax" 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="tServiceForConfDataSet" 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="tServiceConfReportControl" 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="SMV" type="scl:tServiceWithMax" minOccurs="0"/>
    <xs:element name="FileHandling" type="tServiceYesNo" minOccurs="0"/>
    <xs:element name="ConfLNs" type="tConfLNs" minOccurs="0"/>
    <xs:element name="ClientServices" type="tClientServices" minOccurs="0"/>
    <xs:element name="ConfLdName" type="tServiceYesNo" minOccurs="0"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="tAccessPoint">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element name="Server" type="scl:tServer">
            <xs:unique name="uniqueAssociationInServer">
              <xs:selector xpath=".//scl:Association"/>
              <xs:field xpath="@associationID"/>
            </xs:unique>
          </xs:element>
          <xs:element ref="scl:LN" maxOccurs="unbounded"/>
          <xs:element name="ServerAt" type="tServerAt"/>
        </xs:choice>
        <xs:element name="Services" type="scl:tServices" minOccurs="0"/>
        <xs:element name="GOOSESecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
        <xs:element name="SMVSecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
      </xs:sequence>
      <xs:attribute name="name" type="tAccessPointName" use="required"/>
      <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="tCertificate">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="Subject" type="tCert"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:element name="IssuerName" type="tCert"/>
    </xs:sequence>
    <xs:attribute name="xferNumber" type="xs:unsignedInt" use="optional"/>
    <xs:attribute name="serialNumber" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:normalizedString">
                <xs:minLength value="1"/>
                <xs:pattern value="\d+"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tCert">
    <xs:attribute name="commonName" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:normalizedString">
                <xs:minLength value="4"/>
                <xs:pattern value="none"/>
                <xs:pattern value="CN=.+"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="idHierarchy" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:normalizedString">
                <xs:minLength value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="tServerAt">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:attribute name="apName" type="tAccessPointName" use="required"/>
        </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="LDDevice" type="tLDDevice" maxOccurs="unbounded">
                    <xs:unique name="uniqueLNInLDDevice">
                        <xs:selector xpath="/scl:LN"/>
                        <xs:field xpath="@inst"/>
                        <xs:field xpath="@lnClass"/>
                        <xs:field xpath="@prefix"/>
                    </xs:unique>
                    <xs:unique name="uniqueLogName">
                        <xs:selector xpath="/*/scl:Log"/>
                        <xs:field xpath="@name"/>
                    </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="tLDDevice">
    <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="tLDInst" use="required"/>
            <xs:attribute name="ldName" type="tLDName" use="optional"/>
        </xs:extension>
    </xs:complexContent>

```

```

    </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:attributeGroup ref="agLNRef"/>
    <xs:attribute name="kind" type="tAssociationKindEnum" use="required"/>
    <xs:attribute name="associationID" type="tName" use="optional"/>
  </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:unique name="uniqueDOIInLN0">
      <xs:selector xpath="/.scl:DOI"/>
      <xs:field xpath="@name"/>
    </xs:unique>
  </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:unique name="uniqueDOIinLN">
    <xs:selector xpath="/scl:DOI"/>
    <xs:field xpath="@name"/>
  </xs:unique>
  <xs:key name="providedLogName">
    <xs:selector xpath="/scl:Log"/>
    <xs:field xpath="@name"/>
  </xs:key>
</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:unique name="uniqueSDI_DAlinDOI">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="Inputs" type="tInputs" minOccurs="0">
          <xs:unique name="uniqueExtRefInInputs">
            <xs:selector xpath="/scl:ExtRef"/>
            <xs:field xpath="@iedName"/>
            <xs:field xpath="@ldInst"/>
            <xs:field xpath="@prefix"/>
            <xs:field xpath="@lnClass"/>
            <xs:field xpath="@lnInst"/>
            <xs:field xpath="@doName"/>
            <xs:field xpath="@daName"/>
            <xs:field xpath="@intAddr"/>
          </xs:unique>
        </xs:element>
        <xs:element name="Log" type="scl:tLog" minOccurs="0" maxOccurs="unbounded"/>
      </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="tLNInst" use="required"/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
    </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: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>
<xs:complexType name="tDataSet">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:choice maxOccurs="unbounded">
        <xs:element name="FCDA" type="tFCDA"/>
        <xs:element name="FCCB" type="tFCCB"/>
      </xs:choice>
      <xs:attribute name="name" type="tDataSetName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tFCDA">
  <xs:attribute name="ldInst" type="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="optional"/>
  <xs:attribute name="lnInst" type="tLNInst" 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:attribute name="ix" type="xs:unsignedInt" use="optional"/>
</xs:complexType>
<xs:complexType name="tFCCB">
  <xs:attribute name="ldInst" type="tLDInst" use="required"/>
  <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
  <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
  <xs:attribute name="cbName" type="tCBName" use="required"/>
  <xs:attribute name="daName" type="tName" use="optional"/>
  <xs:attribute name="fc" type="tServiceFCEnum" use="required"/>
  <xs:attribute name="mustUnderstand" type="xs:boolean" use="required" fixed="true"/>
</xs:complexType>
<xs:complexType name="tControl" abstract="true">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:attribute name="name" type="tCBName" use="required"/>
      <xs:attribute name="datSet" type="tDataSetName" use="optional"/>
    </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:attribute name="gi" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="includeAll" 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="optional"/>
      <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:attribute name="indexed" type="xs:boolean" use="optional" default="true"/>
    </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="IdInst" type="tLDInst" use="optional"/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
      <xs:attribute name="InClass" type="scl:tLNClassEnum" use="optional" default="LLN0"/>
      <xs:attribute name="InInst" type="tLNInst" use="optional"/>
      <xs:attribute name="logName" type="tLogName" 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="scl:agDesc"/>
  <xs:attribute name="iedName" type="tIEDName" use="optional"/>
  <xs:attribute name="IdInst" type="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="scl:tPrefix" use="optional"/>
  <xs:attribute name="InClass" type="scl:tLNClassEnum" use="optional"/>
  <xs:attribute name="InInst" type="tLNInst" use="optional"/>
  <xs:attribute name="doName" type="scl:tName" use="optional"/>
  <xs:attribute name="daName" type="scl: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:attribute name="name" type="tLogName" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tControlWithIEDName">
  <xs:complexContent>
    <xs:extension base="tControl">
      <xs:sequence>
        <xs:element name="IEDName" type="tIEDName" 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="applID" 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:unique name="uniqueSDI_DAIinSDI">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="DAI" type="tDAI"/>
      </xs:choice>
      <xs:attribute name="name" type="tDataName" 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="tAttributeNameEnum" 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="tAttributeNameEnum" 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="tServiceWithOptionalMax">
  <xs:attribute name="max" type="xs:unsignedInt" use="optional"/>
</xs:complexType>
<xs:complexType name="tServiceWithMax">
  <xs:attribute name="max" type="xs:unsignedInt" use="required"/>
</xs:complexType>
<xs:complexType name="tServiceConfReportControl">
  <xs:complexContent>
    <xs:extension base="tServiceWithMax">
      <xs:attribute name="bufMode" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:Name">
            <xs:enumeration value="unbuffered"/>
            <xs:enumeration value="buffered"/>
            <xs:enumeration value="both"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="bufConf" type="xs:boolean" use="optional" default="false"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</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="tServiceForConfDataSet">
    <xs:complexContent>
        <xs:extension base="tServiceWithMaxAndMaxAttributes">
            <xs:attribute name="modify" type="xs:boolean" use="optional" default="true"/>
            <xs:attribute name="fccb" type="xs:boolean" use="optional" default="false"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tClientServices">
    <xs:attribute name="goose" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="gsse" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="bufReport" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="unbufReport" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="readLog" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="sv" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="fccb" type="xs:boolean" use="optional" default="false"/>
</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="applID" 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:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </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="tConfLNns">
    <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:keyref name="ref2LDeviceInDataSetForFCDAinLN" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet/scl:FCDA"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <xs:keyref name="ref2LDeviceInDataSetForFCCBinLN" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet/scl:FCCB"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <xs:keyref name="ref2LDeviceInDataSetForFCDAinLN0" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet/scl:FCDA"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <xs:keyref name="ref2LDeviceInDataSetForFCCBinLN0" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet/scl:FCCB"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <!--<xs:keyref name="ref2LDeviceInDataSetForFCDAinLN" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <xs:keyref name="ref2LDeviceInDataSetForFCDAinLN0" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref>
    <xs:keyref name="ref2LDeviceInDataSetForFCDAinLN2" refer="LDeviceInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:LN/scl:DataSet"/>
        <xs:field xpath="@IdInst"/>
    </xs:keyref-->
    <xs:key name="AccessPointInIEDKey">
        <xs:selector xpath="/scl:AccessPoint"/>
        <xs:field xpath="@name"/>
    </xs:key>
    <xs:keyref name="ServerAtRef2AccessPoint" refer="AccessPointInIEDKey">
        <xs:selector xpath="/scl:AccessPoint/scl:ServerAt"/>
        <xs:field xpath="@apName"/>
    </xs:keyref>
</xs:element>
</xs:schema>

```

## A.5 Communication subnetworks

### File SCL\_Communication.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:scl="http://www.iec.ch/61850/2003/SCL"
xmlns="http://www.iec.ch/61850/2003/SCL" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="SCL_BaseTypes.xsd"/>
  <xs:complexType name="tControlBlock" abstract="true">
    <xs:complexContent>
      <xs:extension base="tUnNaming">
        <xs:sequence>
          <xs:element name="Address" type="tAddress" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="IdInst" type="tLDInst" use="required"/>
        <xs:attribute name="cbName" type="tCBName" 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:field xpath="@IdInst"/>
            </xs:unique>
            <xs:unique name="uniqueSMVInConnectedAP">
              <xs:selector xpath="/scl:SMV"/>
              <xs:field xpath="@cbName"/>
              <xs:field xpath="@IdInst"/>
            </xs:unique>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
      </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="tIEDName" use="required"/>
        <xs:attribute name="apName" type="tAcsiName" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

</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: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: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: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: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: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: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: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: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: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: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: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: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}"/>
      <xs:attribute name="type" type="tPTypeEnum" use="required" fixed="MAC-Address"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="tP_APPID">
  <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: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:restriction>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="tP_VLAN-ID">
      <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 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" finalDefault="extension" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
Release 2007/10/18.</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:attribute name="version" type="tScIVersion" use="required" fixed="2007"/>
          <xs:attribute name="revision" type="tScIRevision" use="required" fixed="A"/>
        </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:unique name="uniqueLDName">
      <xs:selector xpath="//scl:LDevice"/>
      <xs:field xpath="@ldName"/>
    </xs:unique>
    -->
    <xs:unique name="uniqueLNode">
      <xs:selector xpath="//scl:LNode"/>

```

```
<xs:field xpath="@InInst"/>
<xs:field xpath="@InClass"/>
<xs:field xpath="@iedName"/>
<xs:field xpath="@IdInst"/>
<xs:field xpath="@prefix"/>
</xs:unique>
<xs:keyref name="refConnectedAP2IED" refer="IEDKey">
  <xs:selector xpath="/scl:Communication/scl:SubNetwork/scl:ConnectedAP"/>
  <xs:field xpath="@iedName"/>
</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" version="2007" revision="A">
  <Header id="Normative Enumerations 2007" nameStructure="IEDName"/>
  <DataTypeTemplates>
    <LNodeType id="Dummy" InClass="LLN0">
      <DO name="Mod" type="myMod"/>
    </LNodeType>
    <DOType id="myMod" cdc="INC">
      <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"/>
      <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel"/>
    </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>
  </DataTypeTemplates>

```

```

    <EnumVal ord="6">Vab</EnumVal>
    <EnumVal ord="7">Vbc</EnumVal>
    <EnumVal ord="8">Vca</EnumVal>
    <EnumVal ord="9">Vother</EnumVal>
    <EnumVal ord="10">Aother</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 Extremely 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>
    <EnumVal ord="17">Polynom 1</EnumVal>
    <EnumVal ord="18">Polynom 2</EnumVal>
    <EnumVal ord="19">Polynom 3</EnumVal>
    <EnumVal ord="20">Polynom 4</EnumVal>
    <EnumVal ord="21">Polynom 5</EnumVal>
    <EnumVal ord="22">Polynom 6</EnumVal>
    <EnumVal ord="23">Polynom 7</EnumVal>
    <EnumVal ord="24">Polynom 8</EnumVal>
    <EnumVal ord="25">Polynom 9</EnumVal>
    <EnumVal ord="26">Polynom 10</EnumVal>
    <EnumVal ord="27">Polynom 11</EnumVal>
    <EnumVal ord="28">Polynom 12</EnumVal>
    <EnumVal ord="29">Polynom 13</EnumVal>
    <EnumVal ord="30">Polynom 14</EnumVal>
    <EnumVal ord="31">Polynom 15</EnumVal>
    <EnumVal ord="32">Polynom 16</EnumVal>
    <EnumVal ord="33">Multiline 1</EnumVal>
    <EnumVal ord="34">Multiline 2</EnumVal>
    <EnumVal ord="35">Multiline 3</EnumVal>
    <EnumVal ord="36">Multiline 4</EnumVal>
    <EnumVal ord="37">Multiline 5</EnumVal>
    <EnumVal ord="38">Multiline 6</EnumVal>
    <EnumVal ord="39">Multiline 7</EnumVal>
    <EnumVal ord="40">Multiline 8</EnumVal>
    <EnumVal ord="41">Multiline 9</EnumVal>
    <EnumVal ord="42">Multiline 10</EnumVal>
    <EnumVal ord="43">Multiline 11</EnumVal>
    <EnumVal ord="44">Multiline 12</EnumVal>
    <EnumVal ord="45">Multiline 13</EnumVal>
    <EnumVal ord="46">Multiline 14</EnumVal>
    <EnumVal ord="47">Multiline 15</EnumVal>
    <EnumVal ord="48">Multiline 16</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>
  </EnumType>

```

```

<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">Watts</EnumVal>
  <EnumVal ord="63">VAr</EnumVal>
  <EnumVal ord="64">phi</EnumVal>
  <EnumVal ord="65">cos(phi)</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>
  <EnumVal ord="75">Hz/s</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="AutoRecSt">
  <EnumVal ord="1">Ready</EnumVal>
  <EnumVal ord="2">InProgress</EnumVal>
  <EnumVal ord="3">Successfull</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="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="FltLoop">
  <EnumVal ord="1">PhaseAtoGround</EnumVal>
  <EnumVal ord="2">PhaseBtoGround</EnumVal>
  <EnumVal ord="3">PhaseCtoGround</EnumVal>
  <EnumVal ord="4">PhaseAtoB</EnumVal>
  <EnumVal ord="5">PhaseBtoC</EnumVal>
  <EnumVal ord="6">PhaseCtoA</EnumVal>
  <EnumVal ord="7">Other</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="Health">
  <EnumVal ord="1">Ok</EnumVal>
  <EnumVal ord="2">Warning</EnumVal>
  <EnumVal ord="3">Alarm</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="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="PmpCtl">
    <EnumVal ord="1">Inactive</EnumVal>
    <EnumVal ord="2">Stage1</EnumVal>
    <EnumVal ord="3">Stage2</EnumVal>
    <EnumVal ord="4">Stage3</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">Disconnecter</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">           2005 version - changed for next version!
  <EnumVal ord="1">3 phase tripping</EnumVal>
  <EnumVal ord="2">1 or 3 phase tripping</EnumVal>
  <EnumVal ord="3">specific</EnumVal>
</EnumType -->
<EnumType id="TrMod">           <!-- also to be used for TrBeh -->
  <EnumVal ord="1">single pole</EnumVal>
  <EnumVal ord="2">undefined</EnumVal>
  <EnumVal ord="3">three pole</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>
<!-- new from 2007 data model -->
<EnumType id="CalcMthd">
  <EnumVal ord="1">PRES</EnumVal>
  <EnumVal ord="2">MIN</EnumVal>
  <EnumVal ord="3">MAX</EnumVal>
  <EnumVal ord="4">TOTMIN</EnumVal>
  <EnumVal ord="5">TOTMAX</EnumVal>
  <EnumVal ord="6">AVG</EnumVal>
  <EnumVal ord="7">SDV</EnumVal>
</EnumType>
</DataTypeTemplates>
</SCL>

```

## Annex C (normative)

### 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.

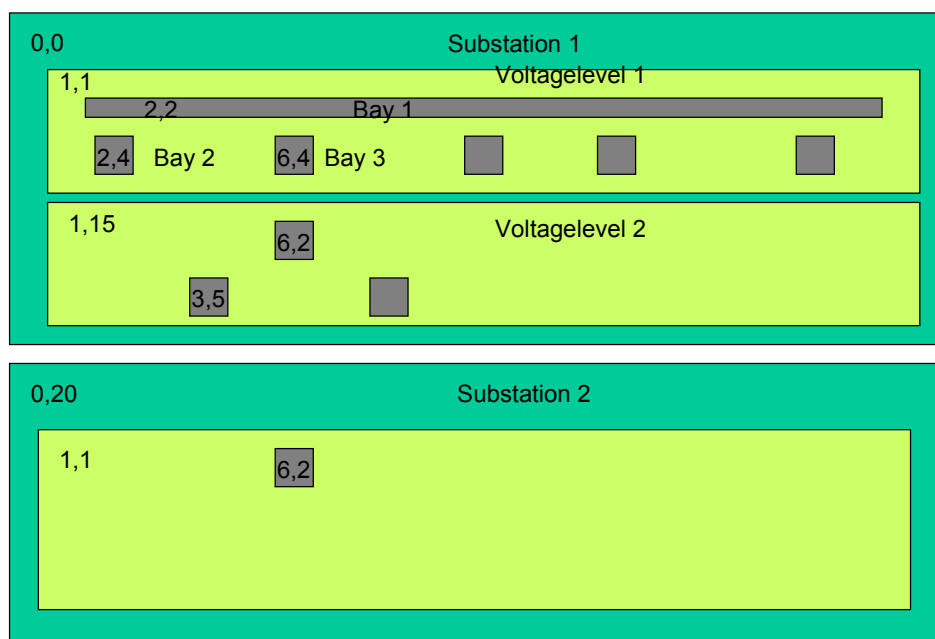


Figure C.1 – Coordinate example

**Additional 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="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      COPYRIGHT IEC, 2005. Version 1.0. Release 2005/09/11.
      This schema is for informational 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 8.3.5).

```
<?xml version="1.0"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sxy="http://www.iec.ch/61850/2003/SCLcoordinates"
  xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL SCL.xsd
    http://www.iec.ch/61850/2003/SCLcoordinates 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 Data model description

This annex defines a schema based on SCL type template definitions for the purpose of formally describing IEC 61850 data models as a base to formally document and maintain the data models of different IEC 61850 application domains, and facilitate automatic checking of IED data models against these definitions. It allows also to document private extensions in a computer understandable way.

This schema allows defining one or more application area models in one or more XML files. Typically there is one file describing CDCs, another one describing LN classes of the application area. However, also small CDC extensions could be contained in the same file as the application specific LN class definitions. For verification tools both together could reside in one file per name space version. Model extensions in new versions of the same name space as well as name spaces of other application areas can reference and inherit from already existing name spaces respective from older versions, however can also replace and thus modify elements of older versions.

The following terms are introduced:

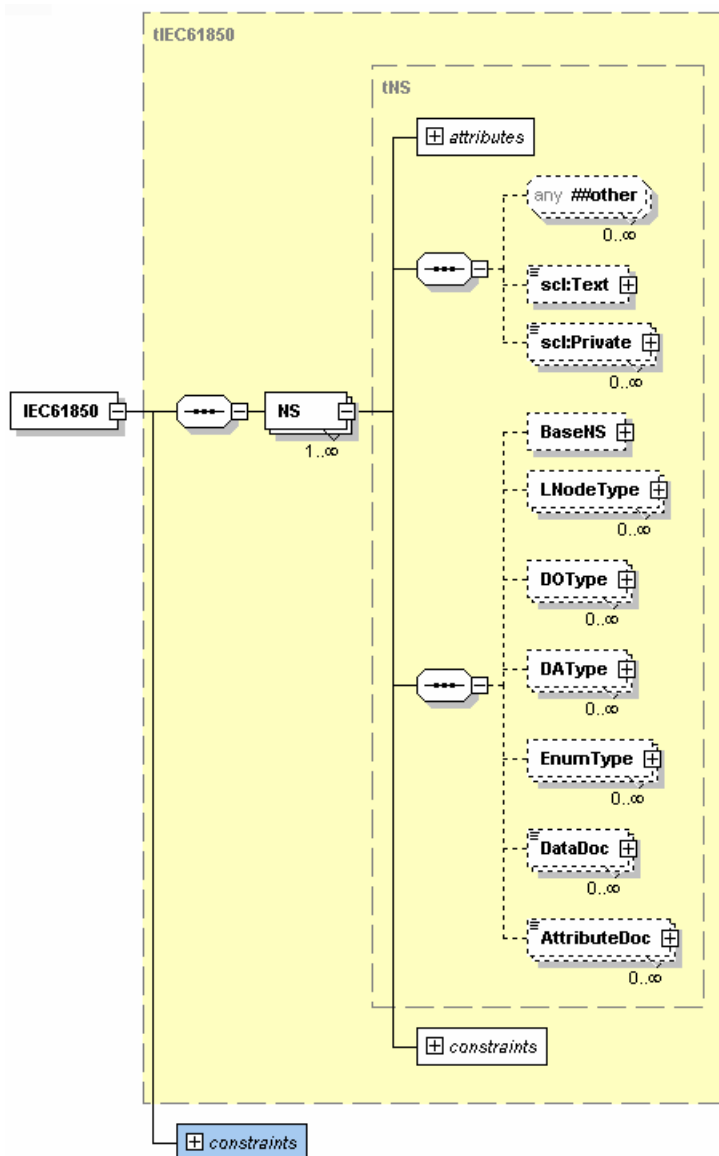
Application area	the application area of the data model. This is some text defining the application area of a name space definition. E.g. the application area of IEC 61850-7-4 is <i>Substation Automation</i> .
Name space	the formal model name space value as introduced in IEC 61850-7-3 at the LPL CDC.
Version	The version of a name space definition. This allows differentiating (upwards) compatible versions for the same name space.
Revision	error solving revisions of the same version, or different drafts of a version before its release. For the usage of version and revision in the context of data models the same general concepts apply as for these terms in the context of the SCL language.

Figure 24 gives an overview on the introduced XML schema. The main element *IEC61850* can contain several name space elements *NS*. The name space element starts with an optional *BaseNS* element, which allows referencing an existing base name space definition, e.g. an existing CDC definition. Its *reuseLNNodeTypes* attribute allows to specify, if only the CDCs of the base NS shall be taken (e.g. if common CDC definitions are reused in another application area), or also the LN class definitions (e.g. in a next version of the same application area). It is followed by *LNodeType* elements, which define the LN classes, and */* or *DOType* elements, which define Common Data Classes, as well as *DAType* elements defining structured attributes. Finally the *EnumType* element allows defining the used enumerations. All these elements are basically syntactically identical to the appropriate elements in the SCL *DataTypeTemplate* section, however have been extended by some attributes allowing to define descriptive text, usage conditions like mandatory and optional, the usage category of DATA or attributes. Further they contain additional *Doc* elements, which allow containing the description part of LN classes respective CDCs as contained in the 7-3 and 7-4 documents, and at the *DOType* element an additional *Services* element, to describe the services allowed on it.

The NS element contains for documentation purpose optional *DataDoc* elements for the description of all DATA as described at the end of IEC61850-7-4, and optional *AttributeDoc* elements to contain the description of all defined attribute names as described at the end of IEC61850-7-3.

As here the *LNodeType* element is used to define LN classes, its *id* attribute shall contain the same value as its *InClass* attribute. Similarly the *DOType* element's *id* attribute shall have in most cases the same value as its *cdc* attribute. Only if CDC variants are used, e.g. due to different enumerations defined in 7-4 for the basic Integer type, then the appropriate CDC variants need a different *id* value, however it shall start with the basic *cdc* attribute value, followed by an underscore.

To clearly see compatible model extensions, the *LNodeType* and *DOType* elements have an additional *extends* attribute, which references the appropriate LN class or CDC in the base name space extended by it. In this case only additions to the extended class are allowed, for forward compatibility all should be optional. If this attribute is missing, and the *I* is identical to one of the *BaseNs* *id*'s, then this means an incompatible redefinition, and all contained elements shall be defined, thus overwriting everything in the base definition.



**Figure 24 Schema overview**

The order of attributes in an implementing data model shall follow the order in the definitions. In case of extensions, the order of attributes shall be first the attributes of the base name space, then followed within each attribute category as defined by the *category* attribute by the newly defined attributes.

The name space for this data model definition schema language shall be the same as for general SCL language: <http://www.iec.ch/61850/2003/SCL>. The SCL\_DataTypeTemplate schema from SCL is reused.

The XML schema definition is:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Extension syntax of SCL for XML description of parts 7-3/7-4.
2007/08/29.</xs:documentation>
  </xs:annotation>
  <!-- <xs:include schemaLocation="SCL_BaseTypes.xsd"/>-->
  <xs:redefine schemaLocation="SCL_DataTypeTemplates.xsd">
    <xs:complexType name="tLNodeType">
      <xs:complexContent>
        <xs:extension base="tLNodeType">
          <xs:sequence>
            <xs:element name="Doc" type="tDoc" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Documentation for this LNodeType.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="agExtension"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDOType">
      <xs:complexContent>
        <xs:extension base="tDOType">
          <xs:sequence>
            <xs:element name="Doc" type="tDoc" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Documentation for this DOType.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="Services" type="xs:string" default="" minOccurs="0"/>
          </xs:sequence>
          <xs:attributeGroup ref="agExtension"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDAType">
      <xs:complexContent>
        <xs:extension base="tDAType">
          <xs:sequence>
            <xs:element name="Doc" type="tDoc" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Documentation for this DAType.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="agExtension"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tEnumType">
      <xs:complexContent>
        <xs:extension base="tEnumType">
          <xs:sequence>
            <xs:element name="Doc" type="tDoc" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Documentation for this EnumType.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="agExtension"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="tDO">
      <xs:complexContent>
```

```

        <xs:extension base="tDO">
            <xs:attributeGroup ref="agCondition"/>
            <xs:attributeGroup ref="agCategory"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSDO">
    <xs:complexContent>
        <xs:extension base="tSDO">
            <xs:attributeGroup ref="agCondition"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tDA">
    <xs:complexContent>
        <xs:extension base="tDA">
            <xs:attributeGroup ref="agCondition"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tBDA">
    <xs:complexContent>
        <xs:extension base="tBDA">
            <xs:attributeGroup ref="agCondition"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:redefine>
<xs:complexType name="tDoc" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:any processContents="lax"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="tDocWithName">
    <xs:complexContent>
        <xs:extension base="tDoc">
            <xs:attribute name="name" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:normalizedString">
                        <xs:minLength value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="tCategory">
    <xs:annotation>
        <xs:documentation>Category a DO belongs to.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:normalizedString">
        <xs:enumeration value="Common"/>
        <xs:enumeration value="Status"/>
        <xs:enumeration value="Controls"/>
        <xs:enumeration value="Measurements"/>
        <xs:enumeration value="Settings"/>
    </xs:restriction>
</xs:simpleType>
<xs:attributeGroup name="agCategory">
    <xs:annotation>
        <xs:documentation>Category description of the element.</xs:documentation>
    </xs:annotation>
    <xs:attribute name="category" use="required">
        <xs:simpleType>
            <xs:restriction base="tCategory"/>
        </xs:simpleType>
    </xs:attribute>
</xs:attributeGroup>
<xs:simpleType name="tCondition">
    <xs:annotation>
        <xs:documentation>Condition on presence</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:normalizedString">
        <xs:minLength value="1"/>
        <xs:enumeration value="M"/>
        <xs:enumeration value="O"/>
        <xs:enumeration value="PICS_SUBST">

```

```

    <xs:annotation>
      <xs:documentation>
        Attribute is mandatory, if substitution is supported (For substitution, see IEC
        61850-7-2).
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_1">
    <xs:annotation>
      <xs:documentation>
        At least one of the attributes shall be present for a given instance of DATA.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_2_1">
    <xs:annotation>
      <xs:documentation>
        All or none of the data attributes belonging to the same group (_1) shall be present for
        a given instance of DATA.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_2_2">
    <xs:annotation>
      <xs:documentation>
        All or none of the data attributes belonging to the same group (_2) shall be present for
        a given instance of DATA.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_2_3">
    <xs:annotation>
      <xs:documentation>
        All or none of the data attributes belonging to the same group (_3) shall be present for
        a given instance of DATA.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_2_4">
    <xs:annotation>
      <xs:documentation>
        All or none of the data attributes belonging to the same group (_4) shall be present for
        a given instance of DATA.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="GC_CON">
    <xs:annotation>
      <xs:documentation>
        A configuration data attribute shall only be present, if the (optional) specific data
        attributes to which this configuration relates, is present as well.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_LN0_M">
    <xs:annotation>
      <xs:documentation>
        The attribute shall be present for LLN0.NamPIt; otherwise (DomainLN.NamPIt) it may be
        optional.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_LN0_EX">
    <xs:annotation>
      <xs:documentation>
        The attribute may be present only if describing LLN0.NamPIt, but not for
        DomainLN.NamPIt.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_DLD_M">
    <xs:annotation>
      <xs:documentation>
        The attribute shall be present if LN name space of this LN (LLN0.NamPIt.InNs or
        Domain.NamPIt.InNs) deviates from the LN name space referenced by IdNs
        (LLN0.NamPIt.IdNs) of the logical device in which this LN is contained.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>

```

```

    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_DLN_M">
    <xs:annotation>
      <xs:documentation>
        The attribute shall be present, if data name space of this data deviates from the data
        name space referenced by either InNs of the logical node in which the data is contained
        or IdNs of the logical device in which the data is contained.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_DLNDA_M">
    <xs:annotation>
      <xs:documentation>
        The attribute shall be present, if CDC name space of this data deviates from the CDC
        name space referenced by either the dataNs of the data, the InNs of the logical node in
        which the data is defined or IdNs of the logical device in which the data is contained.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_SCAV">
    <xs:annotation>
      <xs:documentation>
        The presence of the configuration data attribute depends on the presence of i and f of
        the Analog Value of the data attribute to which this configuration attribute relates.
        For a given data object, that attribute: (1) shall be present, if both i and f are
        present; (2) shall be optional if only i is present, and, (3) is not required if only f
        is present. NOTE: If only i is present in a device without floating point capabilities,
        the configuration parameter may be exchanged offline.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_ST">
    <xs:annotation>
      <xs:documentation>
        If the controllable status class supports status information, the attribute is
        mandatory.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_CO_M">
    <xs:annotation>
      <xs:documentation>
        If the controllable status class supports control, this attribute is available and a
        mandatory attribute.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_CO_O">
    <xs:annotation>
      <xs:documentation>
        If the controllable status class supports control, this attribute is available and an
        optional attribute.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_SG_M">
    <xs:annotation>
      <xs:documentation>
        The attribute is mandatory, if setting group is supported.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_SG_O">
    <xs:annotation>
      <xs:documentation>
        The attribute is optional, if setting group is supported.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="AC_NS_G_M">
    <xs:annotation>
      <xs:documentation>
        The attribute is mandatory, if setting group is not supported.
      </xs:documentation>
    </xs:annotation>
  </xs:enumeration>

```

```

    <xs:enumeration value="AC_NSQ_O">
      <xs:annotation>
        <xs:documentation>
          The attribute is optional, if setting group is not supported.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_RMS_M">
      <xs:annotation>
        <xs:documentation>
          The attribute is mandatory when the harmonics reference type is rms.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_CO_SBO_N_M">
      <xs:annotation>
        <xs:documentation>
          Attribute is mandatory, if the control model SBO with normal security is used.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_CO_SBOW_E_M">
      <xs:annotation>
        <xs:documentation>
          Attribute is mandatory, if the control model SBO with enhanced security is used.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_CO_E_M">
      <xs:annotation>
        <xs:documentation>
          Attribute is mandatory, if control models with enhanced security are used.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_CO_TA_E_M">
      <xs:annotation>
        <xs:documentation>
          Attribute is mandatory, if time activated control and enhanced security are used.
        </xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="AC_CO_SBO_N_M_SBOW_E_M_TA_E_M">
      <xs:annotation>
        <xs:documentation>AC_CO_SBO_N_M and AC_CO_SBOW_E_M and
AC_CO_TA_E_M.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
<xs:attributeGroup name="agCondition">
  <xs:annotation>
    <xs:documentation>Presence condition of the element.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="cond" type="tCondition" use="optional" default="M"/>
</xs:attributeGroup>
<xs:simpleType name="tNSName">
  <xs:annotation>
    <xs:documentation>IEC 61850 Namespace name type</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:attributeGroup name="agNSRef">
  <xs:annotation>
    <xs:documentation>Reference to the default IEC 61850 Namespace that defines the DOTypes, DATypes, and
EnumTypes. These elements shall only be included if they differ from the default settings provided at parent NS
level.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="fromNS" type="tNSName" use="optional" default="IEC 61850-7-4:2003"/>
  <xs:attribute name="version" type="tSciVersion" use="optional" default="2003"/>
  <xs:attribute name="revision" type="tSciRevision" use="optional" default="A"/>
</xs:attributeGroup>
<xs:attributeGroup name="agExtension">
  <xs:annotation>
    <xs:documentation>If true, extends element from base namespace.</xs:documentation>
  </xs:annotation>

```

```

</xs:annotation>
<xs:attribute name="extends" type="xs:boolean" use="optional" default="false"/>
</xs:attributeGroup>
<xs:complexType name="tIEC61850">
  <xs:sequence>
    <xs:element name="NS" type="tNS" maxOccurs="unbounded">
      <xs:unique name="uniqueLNodeTypeNS">
        <xs:selector xpath="scl:LNodeType"/>
        <xs:field xpath="@id"/>
      </xs:unique>
      <xs:unique name="uniqueDOTypeNS">
        <xs:selector xpath="scl:DOType"/>
        <xs:field xpath="@id"/>
      </xs:unique>
      <xs:unique name="uniqueDATypeNS">
        <xs:selector xpath="scl:DAType"/>
        <xs:field xpath="@id"/>
      </xs:unique>
      <xs:unique name="uniqueEnumTypeNS">
        <xs:selector xpath="scl:EnumType"/>
        <xs:field xpath="@id"/>
      </xs:unique>
      <xs:unique name="uniqueDataDoc">
        <xs:selector xpath="scl:DataDoc"/>
        <xs:field xpath="@name"/>
      </xs:unique>
      <xs:unique name="uniqueAttributeDoc">
        <xs:selector xpath="scl:AttributeDoc"/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tNS">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="BaseNS" minOccurs="0">
          <xs:complexType>
            <xs:attributeGroup ref="agNSRef">
              <xs:annotation>
                <xs:documentation>Reference to the default IEC 61850 Namespace that defines the
                DTypes, DATypes, and EnumTypes. These elements shall only be included if they differ from the default settings provided at
                parent NS level.</xs:documentation>
              </xs:annotation>
            </xs:attributeGroup>
            <xs:attribute name="reuseLNodeTypes" type="xs:boolean" use="optional" default="false"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="LNodeType" type="tLNodeType" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueDOInLNodeTypeNS">
            <xs:selector xpath="scl:DO"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="DOType" type="tDOType" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueDAorSDOInDOTypeNS">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
          </xs:unique>
        </xs:element>
        <xs:element name="DAType" type="tDAType" minOccurs="0" maxOccurs="unbounded">
          <xs:unique name="uniqueBDAInDATypeNS">
            <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="uniqueOrdInEnumTypeNS">
            <xs:selector xpath="scl:EnumVal"/>
            <xs:field xpath="@ord"/>
          </xs:unique>
        </xs:element>
        <xs:element name="DataDoc" type="tDocWithName" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="AttributeDoc" type="tDocWithName" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="tScIVersion" use="optional" default="2003"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    <xs:attribute name="revision" type="tSciRevision" use="optional" default="A"/>
    <xs:attribute name="application" use="optional" default="substation automation">
      <xs:simpleType>
        <xs:restriction base="xs:normalizedString">
          <xs:minLength value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="IEC61850" type="tIEC61850">
  <xs:unique name="uniqueNS">
    <xs:selector xpath="/.scl:NS"/>
    <xs:field xpath="@name"/>
  </xs:unique>
</xs:element>
</xs:schema>

```

The following is a small, incomplete example how the schema is used. It defines some LN classes as well as CDCs from the name space IEC 61850-7-4:2003. Observe the usage of CDC variants for different enumerations in the LN class definitions.

```

<?xml version="1.0" encoding="UTF-8"?>
<IEC61850 xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL SCL_Namespaces.xsd"
xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NS name="IEC 61850-7-4:2003" version="2003" revision="A" application="Substation Automation">
    <LNNodeType id="LLN0" InClass="LLN0">
      <DO name="Mod" type="INC_ModeBehaviour" cond="M"/>
      <DO name="Beh" type="INS_ModeBehaviour" cond="M"/>
      <DO name="Health" type="INS_HealthState" cond="M"/>
      <DO name="NamPit" type="LPL" cond="M"/>
      <DO name="Loc" type="SPS" cond="M"/>
      <DO name="OpTmh" type="INS_INT32" cond="O"/>
      <DO name="Diag" type="SPC" cond="O" category="Control"/>
      <DO name="LEDRs" type="SPC" transient="true" cond="O" category="Control"/>
    </LNNodeType>
    <LNNodeType id="XCBR" InClass="XCBR" desc="Circuit breaker">
      <DO name="Mod" type="INC_ModeBehaviour" cond="M"/>
      <DO name="Beh" type="INS_ModeBehaviour" cond="M"/>
      <DO name="Health" type="INS_HealthState" cond="M"/>
      <DO name="NamPit" type="LPL" cond="M"/>
      <DO name="Loc" type="SPS" cond="M"/>
      <DO name="EEHealth" type="INS_HealthState" cond="O"/>
      <DO name="EEName" type="DPL" cond="O"/>
      <DO name="OpCnt" type="INS_INT32" cond="M"/>
      <DO name="Pos" type="DPC" cond="M" category="Control"/>
      <DO name="BlkOpn" type="SPC" cond="M" category="Control"/>
      <DO name="BlkCls" type="SPC" cond="M" category="Control"/>
      <DO name="ChaMotEna" type="SPC" cond="O" category="Control"/>
      <DO name="SumSwARs" type="BCR" cond="O" category="Measurement"/>
      <DO name="CBOpCap" type="INS_BreakerOperatingCapability" cond="M" category="Status"/>
      <DO name="POWCap" type="INS_POWSwitchingCapability" cond="O" category="Status"/>
      <DO name="MaxOpCap" type="INS_BreakerOperatingCapability" cond="O" category="Status"/>
      <Doc>This LN is used for modelling switches with short circuit breaking capability. Additional LNs, for
example SIMG, may be required to complete the logical modelling for the breaker being represented. The closing
and opening commands shall be subscribed from CSWI or CPOW if applicable. If no services with real-time
capability are available between CSWI or CPOW and XCBR, the opening and closing commands are performed with
a GSE-message (see IEC 61850-7-2).</Doc>
    </LNNodeType>
    <DOType cdc="DPC" id="DPC" desc="Controllable double point CDC.">
      <DA name="ctlVal" bType="BOOLEAN" fc="CO" cond="AC_CO_M" category="Control and Status"/>
      <DA name="operTm" bType="Timestamp" fc="CO" cond="AC_CO_O" category="Control and Status"/>
      <DA name="origin" bType="Struct" type="Originator" fc="ST" cond="AC_CO_O" desc="Information
related to the originator of the last change of the controllable data value." category="Control and Status"/>
      <DA name="ctlNum" bType="INT8U" fc="ST" cond="AC_CO_O" desc="0..255" category="Control and
Status"/>
      <DA name="stVal" bType="Dbpos" fc="ST" dchg="true" desc="intermediate-state | off | on | bad-state"
category="Control and Status"/>
      <DA name="q" bType="Quality" fc="ST" qchg="true" desc="Quality of the data value: 'stVal'."
category="Control and Status"/>
      <DA name="t" bType="Timestamp" fc="ST" desc="Timestamp of the last change in one of the attributes
representing the data value ('stVal') or the data quality ('q'), i.e., those which have 'dchg' or 'qchg'."
category="Control and Status"/>

```

```

    <DA name="stSeld" bType="BOOLEAN" fc="ST" dchg="true" cond="AC_CO_O" category="Control and
Status"/>
    <DA name="subEna" bType="BOOLEAN" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subVal" bType="Dbpos" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subQ" bType="Quality" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subID" bType="VisString64" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="pulseConfig" bType="Struct" type="PulseConfig" fc="CF" cond="AC_CO_O"
category="Configuration, Description, and Extension"/>
    <DA name="ctlModel" bType="Enum" type="ctlModel" fc="CF" category="Configuration, Description, and
Extension"/>
    <DA name="sboTimeout" bType="INT32U" fc="CF" cond="AC_CO_O" category="Configuration,
Description, and Extension"/>
    <DA name="sboClass" bType="Enum" type="sboClass" fc="CF" cond="AC_CO_O"
category="Configuration, Description, and Extension"/>
    <DA name="d" bType="VisString255" fc="DC" cond="O" category="Configuration, Description, and
Extension"/>
    <DA name="dU" bType="Unicode255" fc="DC" cond="O" category="Configuration, Description, and
Extension"/>
    <DA name="cdcNs" bType="VisString255" fc="EX" cond="AC_DLNDA_M" category="Configuration,
Description, and Extension"/>
    <DA name="cdcName" bType="VisString255" fc="EX" cond="AC_DLNDA_M" category="Configuration,
Description, and Extension"/>
    <DA name="dataNs" bType="VisString255" fc="EX" cond="AC_DLN_M" category="Configuration,
Description, and Extension"/>
  </DOType>
  <DOType cdc="INS" id="INS_HealthState" desc="Possible values for 'stVal', 'subVal' are restricted by
enumeration. Used for: EEHealth, Health, PhyHealth.">
    <DA name="stVal" bType="Enum" type="Health" fc="ST" dchg="true"/>
    <DA name="q" bType="Quality" fc="ST" qchg="true" desc="Quality of the data value: 'stVal'."/>
    <DA name="t" bType="Timestamp" fc="ST"/>
    <DA name="subEna" bType="BOOLEAN" fc="SV" cond="PICS_SUBST"/>
    <DA name="subVal" bType="Enum" type="Health" fc="SV" cond="PICS_SUBST"/>
    <DA name="subQ" bType="Quality" fc="SV" cond="PICS_SUBST"/>
    <DA name="subID" bType="VisString64" fc="SV" cond="PICS_SUBST"/>
    <DA name="d" bType="VisString255" fc="DC" cond="O"/>
    <DA name="dU" bType="Unicode255" fc="DC" cond="O"/>
    <DA name="cdcNs" bType="VisString255" fc="EX" cond="AC_DLNDA_M"/>
    <DA name="cdcName" bType="VisString255" fc="EX" cond="AC_DLNDA_M"/>
    <DA name="dataNs" bType="VisString255" fc="EX" cond="AC_DLN_M"/>
  </DOType>
  <DAType id="Originator">
    <BDA name="orCat" cond="M" bType="Enum" type="orCategory" desc="Originator category indicates
who/what caused the change of a controllable value. See OriginatorCategory."/>
    <BDA name="orIdnet" cond="M" bType="Octet64" desc="Originator identification shall show the address
of the originator who caused the change of the value. If NULL, originator of a particular action is not known or is not
reported."/>
  </DAType>
  <DataDoc name="Pos">This Data is accessed when performing a switch command or to verify the switch
status or position. When this Data is also used for a hand-operated switch, the (optional) CtlVal attribute in IEC
61850-7-3 does not exist.</DataDoc>
  <DataDoc name="Mod">
    This information reflects the state of the logical node related HW and SW. More detailed information
related to the source of the problem may be provided by specific Data. For LLN0, this Data reflects the worst value
of "Health" of the logical nodes that are part of the logical device associated with LLN0.
    Health states 1 ("green") and 3 ("red") are unambiguous by definition. The detailed meaning of Health
state 2 ("yellow") is a local issue depending from the dedicated function/device.
  </DataDoc>
  <AttributeDoc name="IdNs">Logical device name space. For details see IEC 61850-7-1.</AttributeDoc>
  <AttributeDoc name="InNs">Logical node name space. For details see IEC 61850-7-1.</AttributeDoc>
  <AttributeDoc name="cdcNs">Common data class name space. For details see IEC 61850-7-
1.</AttributeDoc>
</NS>
<NS name="IEC 61850-7-4:2007" version="2003" revision="A" application="substation automation">
  <BaseNS fromNS="IEC 61850-7-4:2003" version="2003" revision="A" reuseLNodeTypes="true"/>
  <DOType cdc="DPC" id="DPC" desc="Controllable double point CDC." extends="false">
    <DA name="stVal" bType="Dbpos" fc="ST" dchg="true" desc="intermediate-state | off | on | bad-state"
category="Status"/>
    <DA name="q" bType="Quality" fc="ST" qchg="true" desc="Quality of the data value: 'stVal'."
category="Status"/>
    <DA name="t" bType="Timestamp" fc="ST" desc="Timestamp of the last change in one of the attributes
representing the data value ('stVal') or the data quality ('q'), i.e., those which have 'dchg' or 'qchg'."
category="Status"/>
    <DA name="stSeld" bType="BOOLEAN" fc="ST" dchg="true" cond="AC_CO_O" category="Status"/>
    <DA name="subEna" bType="BOOLEAN" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subVal" bType="Dbpos" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subQ" bType="Quality" fc="SV" cond="PICS_SUBST" category="Substitution"/>
    <DA name="subID" bType="VisString64" fc="SV" cond="PICS_SUBST" category="Substitution"/>

```

```
<DA name="pulseConfig" bType="Struct" type="PulseConfig" fc="CF" cond="AC_CO_O"
category="Configuration, Description, and Extension"/>
<DA name="ctlModel" bType="Enum" type="ctlModel" fc="CF" category="Configuration, Description, and
Extension"/>
<DA name="sboTimeout" bType="INT32U" fc="CF" cond="AC_CO_O" category="Configuration,
Description, and Extension"/>
<DA name="sboClass" bType="Enum" type="sboClass" fc="CF" cond="AC_CO_O"
category="Configuration, Description, and Extension"/>
<DA name="d" bType="VisString255" fc="DC" cond="O" category="Configuration, Description, and
Extension"/>
<DA name="dU" bType="Unicode255" fc="DC" cond="O" category="Configuration, Description, and
Extension"/>
<DA name="cdcNs" bType="VisString255" fc="EX" cond="AC_DLNDA_M" category="Configuration,
Description, and Extension"/>
<DA name="cdcName" bType="VisString255" fc="EX" cond="AC_DLNDA_M" category="Configuration,
Description, and Extension"/>
<DA name="dataNs" bType="VisString255" fc="EX" cond="AC_DLN_M" category="Configuration,
Description, and Extension"/>
</DOType>
</NS>
</IEC61850>
```

## 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

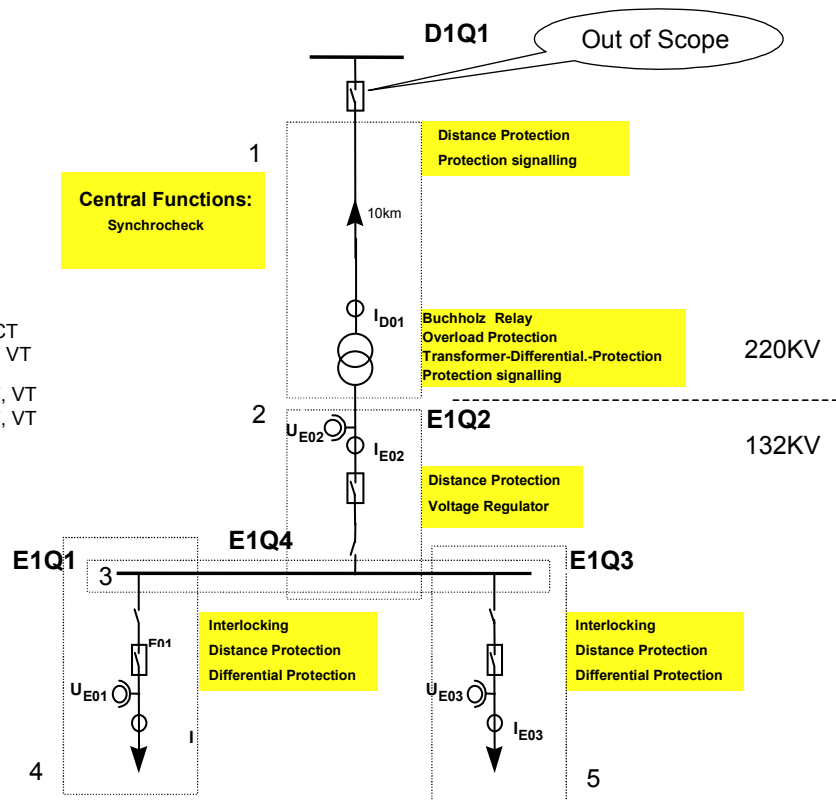
### Example T 1-1

2 Voltage Levels

D1 – 220 kV  
E1 – 132 kV

5 Bays

- 1 – D1Q1 Feeder with Transformer, CT
- 2 – E1Q2 Feeder with DIS, CBR, CT, VT
- 3 – E1Q4 Static Busbar
- 4 – E1Q1 Feeder with DIS, CBR, CT, VT
- 5 – E1Q3 Feeder with DIS, CBR, CT, VT



**Figure D.1 – T1-1 Substation configuration**

Figure D.1 shows the single line. The current infeed 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.

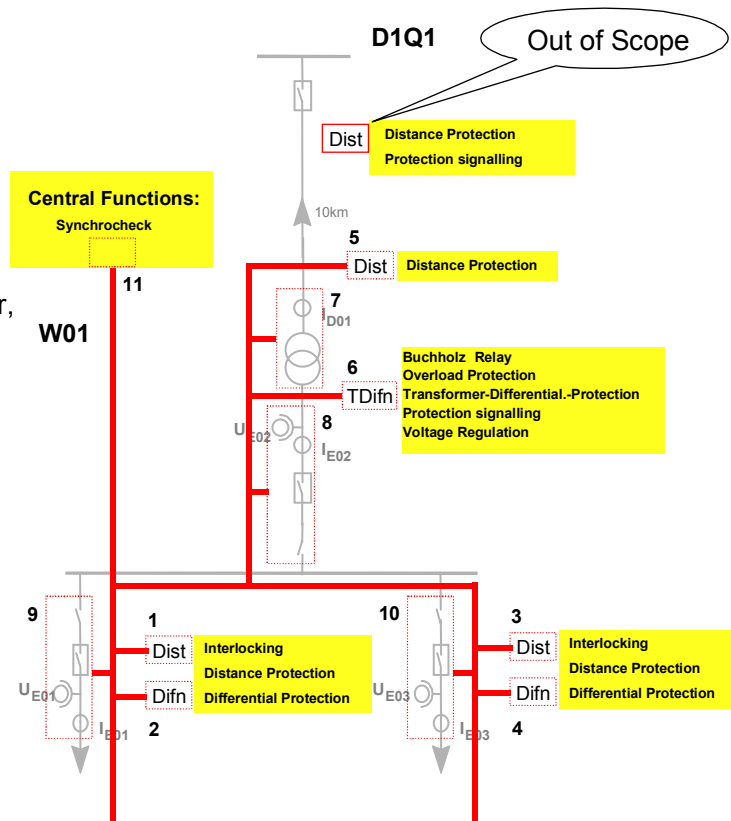
**D.1.2 Communication system configuration**

**Example T 1-1**

Single communication bus

IEDs for:  
 Transformer.  
 Combined Bay Unit (Circuit Breaker,  
 Disconnecter, CT and VT).  
 Each Protection.  
 Central Functions.

No. Name	ID
1 Dist	E1Q1BP3 (PDIS)
2 Difn	E1Q1BP2 (PDIF)
3 Dist	E1Q3BP3 (PDIS)
4 Difn	E1Q3BP2 (PDIF)
5 Dist	D1Q1BP3 (PDIS)
6 TDifn	D1Q1BP2 (PDIF)
7 Trafo	D1Q1SB1
8 LV Bay1	E1Q2SB1
9 LV Bay2	E1Q1SB1
10 LV Bay3	E1Q3SB1
11 Central	D1Q1SB4 (CILO, RSYN)



**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 the instantiated functionality for the transformer control IED as logical nodes.

# Example T 1-1

Single communications bus

IED for: Transformer bay.

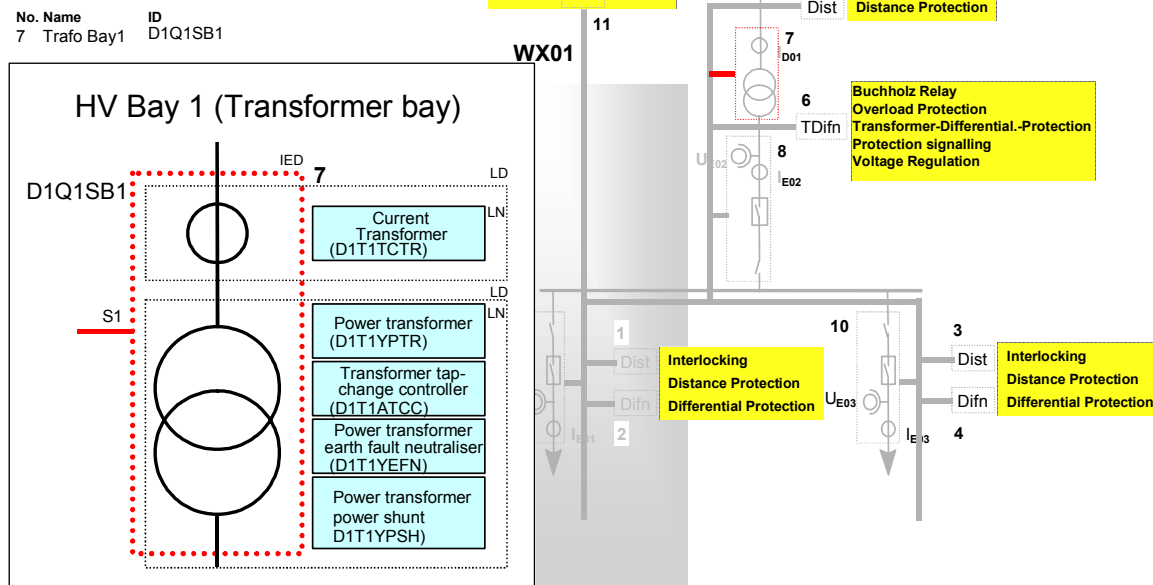


Figure D.3 – T1-1 Transformer bay

## 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:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.iec.ch/61850/2003/SCL" version="2007"
revision="A">
  <Header id="SCL Example T1-1" toolID="MyTool" nameStructure="IEDName" />
  <Substation name="S12" desc="Baden">
    <PowerTransformer name="T1" type="PTR">
      <LNNode lnInst="1" lnClass="PDIF" idInst="F1" iedName="D1Q1BP2" />
      <LNNode lnInst="1" lnClass="TCTR" idInst="C1" iedName="D1Q1SB1" />
      <LNNode lnInst="1" lnClass="YLTC" idInst="S12D1T1" iedName="None" />
      <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>
    <VoltageLevel name="D1">
      <Voltage multiplier="k" unit="V">220</Voltage>
      <Bay name="Q1">
        <LNNode iedName="none" idInst="S12D1Q1" lnClass="PDIS" lnInst="1" />
        <ConductingEquipment name="11" type="CTR">
          <Terminal connectivityNode="S12/D1/Q1/L1" substationName="S12" voltageLevelName="D1" bayName="Q1"
cNodeName="L1" />
          <SubEquipment name="R" phase="A">
```

```

    <LNode iedName="D1Q1BP2" IdInst="F1" InClass="TCTR" InInst="1" />
  </SubEquipment>
  <SubEquipment name="S" phase="B">
    <LNode iedName="D1Q1BP2" IdInst="F1" InClass="TCTR" InInst="2" />
  </SubEquipment>
  <SubEquipment name="T" phase="C">
    <LNode iedName="D1Q1BP2" IdInst="F1" InClass="TCTR" InInst="3" />
  </SubEquipment>
  <SubEquipment name="IO" phase="N">
    <LNode iedName="D1Q1BP2" IdInst="F1" InClass="TCTR" InInst="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 iedName="E1Q1SB1" IdInst="C1" InClass="MMXN" InInst="1" />
    <LNode iedName="none" IdInst="S12E1Q1" InClass="PDIS" InInst="1" />
    <LNode iedName="none" IdInst="F1" InClass="PDIF" InInst="1" />
    <ConductingEquipment name="QA1" type="CBR">
      <LNode iedName="E1Q1SB1" IdInst="C1" InClass="CSWI" InInst="1" />
      <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">
      <LNode iedName="E1Q1SB1" IdInst="C1" InClass="CSWI" InInst="2" />
      <LNode iedName="D1Q1SB4" IdInst="C1" InClass="CILO" InInst="1" />
      <Terminal connectivityNode="S12/E1/Q4/B1" substationName="S12" voltageLevelName="E1" bayName="Q4"
cnodeName="B1" />
      <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" />
    <SubEquipment name="A" phase="A">
      <LNode iedName="E1Q1SB1" IdInst="C1" InClass="TVTR" InInst="1" />
    </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/L2" substationName="S12" voltageLevelName="E1" bayName="Q1"
cnodeName="L2" />
    </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" />
  </Bay>
  <Bay name="Q2" desc="Turgi">
    <ConductingEquipment name="QA1" type="CBR">
      <LNode iedName="D1Q1SB4" IdInst="C1" InClass="CSWI" InInst="1" />
      <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 iedName="D1Q1SB4" IdInst="C1" InClass="CSWI" InInst="2" />
      <LNode iedName="D1Q1SB4" IdInst="C1" InClass="CILO" InInst="2" />
      <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/L1" substationName="S12" voltageLevelName="E1" bayName="Q2"
cNodeName="L1" />
  </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" />
</Bay>
<Bay name="Q3" desc="London">
  <LNode iedName="None" IdInst="LD0" InClass="MMXU" InInst="1" />
  <LNode iedName="None" IdInst="LD0" InClass="PDIS" InInst="1" />
  <LNode iedName="None" IdInst="LD0" InClass="PDIF" InInst="1" />
  <ConductingEquipment name="QA1" type="CBR">
    <LNode iedName="None" IdInst="C1" InClass="CSWI" InInst="1" />
    <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/Q4/B1" substationName="S12" voltageLevelName="E1" bayName="Q4"
cNodeName="B1" />
      <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" />
    </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/L2" substationName="S12" voltageLevelName="E1" bayName="Q3"
cNodeName="L2" />
    </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" />
  </Bay>
</Bay name="Q4">
  <ConnectivityNode name="B1" pathName="S12/E1/Q4/B1" />
</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">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>
    <GSE IdInst="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>
      <MinTime unit="s">4</MinTime>
      <MaxTime unit="s">1000</MaxTime>
    </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 IdInst="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 IdInst="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>
<ConnectedAP iedName="A1KA1" apName="S1">
  <Address>
    <P type="IP">10.0.0.121</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="E1Q2SB1" apName="S1" >
  <Address>
    <P type="IP">10.0.0.14</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 />
    <ConfDataSet max="4" maxAttributes="50" />
    <ReadWrite />
    <ConfReportControl max="12" />
    <GetCBValues />
    <ConfLogControl max="1" />
    <ReportSettings cbName="Conf" dataSet="Conf" rptID="Dyn" optFields="Conf" bufTime="Dyn" intgPd="Dyn" />
    <GSESettings cbName="Conf" dataSet="Conf" applID="Conf" />
    <GOOSE max="2" />
    <FileHandling />
  </Services>
</IED>

```

```

    <ConfLNns fixLnInst="true" />
  </Services>
  <AccessPoint name="S1">
    <Server>
      <Authentication none="true" />
      <LDevice inst="C1">
        <LN0 inst="" InClass="LLN0" InType="LN0">
          <DataSet name="Positions">
            <FCDA IdInst="C1" prefix="" InClass="CSWI" InInst="1" doName="Pos" fc="ST" />
            <FCDA IdInst="C1" prefix="" InClass="CSWI" InInst="2" doName="Pos" fc="ST" />
          </DataSet>
          <DataSet name="Measurands">
            <FCDA IdInst="C1" prefix="" InClass="MMXN" InInst="1" doName="Amp" fc="MX" />
            <FCDA IdInst="C1" prefix="" InClass="MMXN" InInst="1" doName="Volt" fc="MX" />
          </DataSet>
          <DataSet name="smv">
            <FCDA IdInst="C1" prefix="" InClass="TVTR" InInst="1" doName="Vol" daName="instMag" fc="MX" />
          </DataSet>
          <ReportControl name="PosReport" dataSet="Positions" confRev="1">
            <TrgOps dchg="true" qchg="true" />
            <RptEnabled max="5">
              <ClientLN iedName="A1KA1" IdInst="LD0" InInst="1" InClass="IHMI" />
            </RptEnabled>
          </ReportControl>
          <ReportControl name="MeaReport" dataSet="Measurands" confRev="1" intgPd="2000">
            <TrgOps qchg="true" period="true" />
            <OptFields reasonCode="true" />
            <RptEnabled max="5">
              <ClientLN iedName="A1KA1" IdInst="none" InInst="1" InClass="IHMI" />
            </RptEnabled>
          </ReportControl>
          <LogControl name="Log" dataSet="Positions" logName="C1">
            <TrgOps dchg="true" qchg="true" />
          </LogControl>
          <Log />
          <GSEControl name="ItlPositions" dataSet="Positions" applID="Itl">
            <IEDName>E1Q2SB1</IEDName>
          </GSEControl>
          <SampledValueControl name="Volt" dataSet="smv" smvID="11" smpRate="4800" nofASDU="5">
            <IEDName>D1Q1SB4</IEDName>
            <SmvOpts refreshTime="true" sampleSynchronized="true" sampleRate="true" />
          </SampledValueControl>
        </LN0>
        <LN inst="1" InClass="LPHD" InType="LPHDa">
          <DOI name="Proxy">
            <DAI name="stVal" valKind="Set">
              <Val>false</Val>
            </DAI>
          </DOI>
        </LN>
        <LN inst="1" InClass="CSWI" InType="CSWIa" />
        <LN inst="2" InClass="CSWI" InType="CSWIa" />
        <LN inst="1" InClass="MMXN" InType="MMXNa">
          <DOI name="Volt">
            <SDI name="sVC">
              <DAI name="scaleFactor" valKind="Set">
                <Val>200</Val>
              </DAI>
              <DAI name="offset" valKind="Set">
                <Val>10</Val>
              </DAI>
            </SDI>
          </DOI>
        </LN>
        <LN inst="1" InClass="TVTR" InType="TVTRa" />
      </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>
<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">
  <Services>
    <DynAssociation />
    <GetDirectory />
    <GetDataObjectDefinition />
    <GetDataSetValue />
    <DataSetDirectory />
    <ConfDataSet max="4" />
    <ReadWrite />
    <ConfReportControl max="12" />
    <GetCBValues />
    <ReportSettings cbName="Conf" dataSet="Conf" rptID="Dyn" optFields="Dyn" bufTime="Dyn" intgPd="Dyn" />
    <GSESettings cbName="Conf" dataSet="Conf" applID="Conf" />
    <GOOSE max="2" />
    <FileHandling />
  </Services>
  <AccessPoint name="S1">
    <Server>
      <Authentication none="true" />
      <LDevice inst="F1">
        <LN0 inst="" InClass="LLN0" InType="LN0" />
        <LN inst="1" InClass="LPHD" InType="LPHDa">
          <DOI name="Proxy">
            <DAI name="stVal" valKind="Set">
              <Val>false</Val>
            </DAI>
          </DOI>
        </LN>
        <LN inst="1" InClass="TCTR" InType="TCTRa" />
        <LN inst="2" InClass="TCTR" InType="TCTRa" />
        <LN inst="3" InClass="TCTR" InType="TCTRa" />
        <LN inst="4" InClass="TCTR" InType="TCTRa" />
      </LDevice>
    </Server>
  </AccessPoint>
</IED>
<IED name="D1Q1BP3">
  <AccessPoint name="S1" />
</IED>
<IED name="D1Q1SB4">
  <Services>
    <DynAssociation />
    <GetDirectory />
    <GetDataObjectDefinition />
    <GetDataSetValue />
    <DataSetDirectory />
    <ConfDataSet max="4" />
    <ReadWrite />
    <ConfReportControl max="12" />
    <GetCBValues />
    <ConfLogControl max="1" />
    <ReportSettings cbName="Conf" dataSet="Conf" rptID="Dyn" optFields="Conf" bufTime="Dyn" intgPd="Dyn" />
    <GSESettings cbName="Conf" dataSet="Conf" applID="Conf" />
    <GOOSE max="2" />
    <FileHandling />
  </Services>
  <AccessPoint name="S1">
    <Server>
      <Authentication none="true" />
      <LDevice inst="C1">
        <LN0 inst="" InClass="LLN0" InType="LN0">

```

```

    <DataSet name="SyckResult">
      <FCDA IdInst="C1" prefix="" InClass="RSYN" InInst="1" doName="Rel" fc="ST" />
    </DataSet>
    <GSEControl name="SyckResult" datSet="SyckResult" confRev="1" appID="SynChk">
      <IEDName>E1Q1SB1</IEDName>
    </GSEControl>
  </LNO>
  <LN inst="1" InClass="LPHD" InType="LPHDa">
    <DOI name="Proxy">
      <DAI name="stVal" valKind="Set">
        <Val>false</Val>
      </DAI>
    </DOI>
  </LN>
  <LN inst="1" InClass="RSYN" InType="RSYNa" />
  <LN inst="1" InClass="CILO" InType="CILOa" />
  <LN inst="2" InClass="CILO" InType="CILOa" />
  <LN inst="1" InClass="CSWI" InType="CSWIa" />
  <LN inst="2" InClass="CSWI" InType="CSWIa" />
</LDevice>
</Server>
</AccessPoint>
</IED>
<IED name="A1KA1">
  <AccessPoint name="S1">
    <LN inst="1" InClass="IHMI" InType="IHMIa" />
  </AccessPoint>
</IED>
<DataTypeTemplates>
  <LNNodeType id="LNO" InClass="LLNO">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myBeh" />
    <DO name="Health" type="myHealth" />
    <DO name="NamPit" type="myLN0LPL" />
  </LNNodeType>
  <LNNodeType id="LPHDa" InClass="LPHD">
    <DO name="PhyNam" type="myDPL" />
    <DO name="PhyHealth" type="myINS" />
    <DO name="Proxy" type="mySPS" />
  </LNNodeType>
  <LNNodeType id="CSWIa" InClass="CSWI">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myBeh" />
    <DO name="Health" type="myHealth" />
    <DO name="NamPit" type="myLPL" />
    <DO name="Pos" type="myPos" />
  </LNNodeType>
  <LNNodeType id="MMXNa" InClass="MMXN">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myHealth" />
    <DO name="Health" type="myBeh" />
    <DO name="NamPit" type="myLPL" />
    <DO name="Amp" type="myMV" />
    <DO name="Vol" type="myMV" />
  </LNNodeType>
  <LNNodeType id="CILOa" InClass="CILO">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myBeh" />
    <DO name="Health" type="myHealth" />
    <DO name="NamPit" type="myLPL" />
    <DO name="EnaOpn" type="mySPS" />
    <DO name="EnaCls" type="mySPS" />
  </LNNodeType>
  <LNNodeType id="TVTRa" InClass="TVTR">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myBeh" />
    <DO name="Health" type="myHealth" />
    <DO name="NamPit" type="myLPL" />
    <DO name="Vol" type="mySAV" />
    <DO name="FuFail" type="mySPS" />
  </LNNodeType>
  <LNNodeType id="TCTRa" InClass="TCTR">
    <DO name="Mod" type="myMod" />
    <DO name="Beh" type="myBeh" />
    <DO name="Health" type="myHealth" />
    <DO name="NamPit" type="myLPL" />
    <DO name="Amp" type="mySAV" />
  </LNNodeType>

```

```

</LNNodeType>
<LNNodeType id="RSYNa" InClass="RSYN">
  <DO name="Mod" type="myMod" />
  <DO name="Beh" type="myBeh" />
  <DO name="Health" type="myHealth" />
  <DO name="NamPlt" type="myLPL" />
  <DO name="Rel" type="mySPS" />
</LNNodeType>
<LNNodeType id="IHMIa" InClass="IHMI">
  <DO name="Mod" type="myMod" />
  <DO name="Beh" type="myBeh" />
  <DO name="Health" type="myHealth" />
  <DO name="NamPlt" type="myLPL" />
</LNNodeType>
<DOType id="myMod" cdc="INC">
  <DA name="stVal" fc="ST" dchg="true" bType="Enum" type="Mod" />
  <DA name="q" fc="ST" qchg="true" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
  <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
  <DA name="Oper" fc="CO" bType="Struct" type="myModOper" />
</DOType>
<DOType id="myHealth" cdc="INS">
  <DA name="stVal" fc="ST" dchg="true" bType="Enum" type="Health" />
  <DA name="q" fc="ST" qchg="true" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOType>
<DOType id="myBeh" cdc="INS">
  <DA name="stVal" fc="ST" dchg="true" bType="Enum" type="Beh" />
  <DA name="q" fc="ST" qchg="true" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOType>
<DOType id="myINS" cdc="INS">
  <DA name="stVal" fc="ST" dchg="true" bType="INT32" />
  <DA name="q" fc="ST" qchg="true" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOType>
<DOType id="myLN0LPL" cdc="LPL">
  <DA name="vendor" fc="DC" bType="VisString255">
    <Val>myVendorName</Val>
  </DA>
  <DA name="swRev" fc="DC" bType="VisString255">
    <Val>my SW revision ID</Val>
  </DA>
  <DA name="d" fc="DC" bType="VisString255" />
  <DA name="configRev" fc="DC" bType="VisString255">
    <Val>Rev 3.45</Val>
  </DA>
  <DA name="ldNs" fc="EX" bType="VisString255">
    <Val>IEC 61850-7-4:2003</Val>
  </DA>
</DOType>
<DOType id="myLPL" cdc="LPL">
  <DA name="vendor" fc="DC" bType="VisString255">
    <Val>myVendorName</Val>
  </DA>
  <DA name="swRev" fc="DC" bType="VisString255" />
  <DA name="d" fc="DC" bType="VisString255" />
</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" dchg="true" bType="Dbpos" />
  <DA name="q" fc="ST" qchg="true" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
  <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
  <DA name="Oper" fc="CO" bType="Struct" type="myOper" />
  <DA name="SBOW" fc="CO" bType="Struct" type="myOper" />
  <DA name="Cancel" fc="CO" bType="Struct" type="myCancel" />
</DOType>
<DOType id="mySPS" cdc="SPS">
  <DA name="stVal" fc="ST" dchg="true" bType="BOOLEAN" />

```

```

    <DA name="q" fc="ST" qchg="true" bType="Quality" />
    <DA name="t" fc="ST" bType="Timestamp" />
  </DOType>
  <DOType id="myMV" cdc="MV">
    <DA name="mag" fc="MX" dchg="true" bType="Struct" type="myAnalogValue" />
    <DA name="q" fc="MX" qchg="true" bType="Quality" />
    <DA name="t" fc="MX" bType="Timestamp" />
    <DA name="sVC" fc="CF" dchg="true" bType="Struct" type="ScaledValueConfig" />
  </DOType>
  <DOType id="myCMV" cdc="CMV">
    <DA name="cVal" fc="MX" dchg="true" bType="Struct" type="myVector" />
    <DA name="q" fc="MX" qchg="true" bType="Quality" />
    <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" qchg="true" bType="Quality" />
  </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>
  <DAType id="originator">
    <BDA name="orCat" bType="Enum" type="orCategory" />
    <BDA name="orIdent" bType="Octet64" />
  </DAType>
  <DAType id="myModOper">
    <BDA name="ctlVal" bType="Enum" type="Mod" />
    <BDA name="origin" bType="Struct" type="originator" />
    <BDA name="ctlNum" bType="INT8U" />
    <BDA name="T" bType="Timestamp" />
    <BDA name="Test" bType="BOOLEAN" />
    <BDA name="Check" bType="Check" />
  </DAType>
  <DAType id="myOper">
    <BDA name="ctlVal" bType="BOOLEAN" />
    <BDA name="origin" bType="Struct" type="originator" />
    <BDA name="ctlNum" bType="INT8U" />
    <BDA name="T" bType="Timestamp" />
    <BDA name="Test" bType="BOOLEAN" />
    <BDA name="Check" bType="Check" />
  </DAType>
  <DAType id="myCancel">
    <BDA name="ctlVal" bType="BOOLEAN" />
    <BDA name="origin" bType="Struct" type="originator" />
    <BDA name="ctlNum" bType="INT8U" />
    <BDA name="T" bType="Timestamp" />
    <BDA name="Test" bType="BOOLEAN" />
  </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>
<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="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="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)

### SCL syntax: General XML schema definition

This annex contains the SCL schema which informally defines everything which shall be syntactically allowed to cover the language versions 2003 A and 2007 A. It has to be kept in mind, that this schema can not be used as input check in general, because it surely fails for any follower SCL version. It should just give an idea what a version 2007 tool has to accept as input, beneath the usage of mustUnderstand and mayIignore rules.

#### E.1 Base types

##### SCL\_BaseSimpleTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.iec.ch/61850/2003/SCL"
targetNamespace="http://www.iec.ch/61850/2003/SCL" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">SCL informative schema. Version 2.0. (SCL language version "2003"). Release
2007/10/18.</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="tAcsiName">
    <xs:restriction base="xs:Name">
      <xs:pattern value="\p{L}[\d,\p{L},_]*"/>
    </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{L}[\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:simpleType name="tScIVersion">
    <xs:restriction base="tName">
      <xs:pattern value="20\d{2}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tScIRevision">
    <xs:restriction base="xs:Name">
      <xs:pattern value="\p{Lu}"/>
    </xs:restriction>
  </xs:simpleType>
```

```

    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tRestrLdName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tEmpty">
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tIEDName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLDName">
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="64"/>
      <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLDInst">
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="64"/>
      <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLDInstOrEmpty">
    <xs:union memberTypes="tLDInst tEmpty"/>
  </xs:simpleType>
  <xs:simpleType name="tPrefix">
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="6"/>
      <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
      <xs:pattern value=""/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLNInstOrEmpty">
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="7"/>
      <xs:pattern value="[d]*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLNInst">
    <xs:restriction base="tLNInstOrEmpty">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tDataName">
    <xs:restriction base="tRestrName1stU">
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tDataSetName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tCBName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tLogName">
    <xs:restriction base="tAcsiName">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tAccessPointName">
    <xs:restriction base="xs:normalizedString">
      <xs:pattern value="[d,\p{L}][d,\p{L},_]*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

**SCL\_Enums.xsd**

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">SCL informative schema. Version 2.0. (SCL language version "2003"). Release
2007/10/18</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}{1,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:enumeration value="Oper"/>
      <xs:enumeration value="SBO"/>
      <xs:enumeration value="SBOw"/>
      <xs:enumeration value="Cancel"/>
    </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:restriction>
  </xs:simpleType>

```

```

        <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:enumeration value="AB" />
        <xs:enumeration value="BC" />
        <xs:enumeration value="CA" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tAuthenticationEnum">
    <xs:restriction base="xs:Name">
        <xs:enumeration value="none" />
        <xs:enumeration value="password" />
        <xs:enumeration value="weak" />
        <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="[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="PFRG"/>
      <xs:enumeration value="PHAR"/>
      <xs:enumeration value="PHIZ"/>
      <xs:enumeration value="PIOC"/>
      <xs:enumeration value="PMRI"/>
    </xs:restriction>
  </xs:simpleType>

```

```

    <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="PTRR"/>
    <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:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="ZGIL"/>
    <xs:enumeration value="ZLIN"/>
    <xs:enumeration value="ZMOT"/>
    <xs:enumeration value="ZREA"/>
    <xs:enumeration value="ZRRR"/>
    <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="tServiceFCEnum">
  <xs:restriction base="xs:Name">
    <xs:enumeration value="SG"/>
    <xs:enumeration value="BR"/>
    <xs:enumeration value="RP"/>
    <xs:enumeration value="LG"/>
    <xs:enumeration value="GO"/>
    <xs:enumeration value="GS"/>
    <xs:enumeration value="MS"/>
    <xs:enumeration value="US"/>
  </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="VisString129"/>
    <xs:enumeration value="VisString255"/>
    <xs:enumeration value="Octet64"/>
    <xs:enumeration value="Unicode255"/>
    <xs:enumeration value="Struct"/>
    <xs:enumeration value="EntryTime"/>
    <xs:enumeration value="Check"/>
    <xs:enumeration value="ObjRef"/>
  </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:restriction>

```

```

        <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="m2"/>
        <xs:enumeration value="m3"/>
        <xs:enumeration value="m/s"/>
        <xs:enumeration value="m/s2"/>
        <xs:enumeration value="m3/s"/>
        <xs:enumeration value="m/m3"/>
        <xs:enumeration value="M"/>
        <xs:enumeration value="kg/m3"/>
        <xs:enumeration value="m2/s"/>
        <xs:enumeration value="W/m K"/>
        <xs:enumeration value="J/K"/>
        <xs:enumeration value="ppm"/>
        <xs:enumeration value="1/s"/>
        <xs:enumeration value="rad/s"/>
        <xs:enumeration value="VA"/>
        <xs:enumeration value="VAR"/>
        <xs:enumeration value="phi"/>
        <xs:enumeration value="cos_phi"/>
        <xs:enumeration value="Vs"/>
        <xs:enumeration value="V2"/>
        <xs:enumeration value="As"/>
        <xs:enumeration value="A2"/>
        <xs:enumeration value="VAh"/>
        <xs:enumeration value="Wh"/>
        <xs:enumeration value="VArh"/>
        <xs:enumeration value="V/Hz"/>
        <xs:enumeration value="Hz/s"/>
        <xs:enumeration value="char"/>
        <xs:enumeration value="char/s"/>
        <xs:enumeration value="kgm2"/>
        <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:simpleType name="tRightEnum">
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="full"/>
    <xs:enumeration value="fix"/>
    <xs:enumeration value="dataflow"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## SCL\_BaseTypes.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">SCL informative schema. Version 2.0. (SCL language version "2003"). Release
    2007/10/18.</xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="SCL_Enums.xsd"/>
  <xs:attributeGroup name="agDesc">
    <xs:attribute name="desc" type="xs:normalizedString" use="optional" default=""/>
  </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="tAcsiNaming" abstract="true">
    <xs:complexContent>
        <xs:extension base="tBaseElement">
            <xs:attribute name="name" type="tAcsiName" use="required"/>
            <xs:attributeGroup ref="agDesc"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="tAnyContentFromOtherNamespace" abstract="true" mixed="true">
    <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: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:complexContent mixed="true">
        <xs:extension base="tAnyContentFromOtherNamespace">
            <xs:attribute name="type" type="xs:normalizedString" use="required"/>
            <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" default=""/>
    <xs:attribute name="toolID" type="xs:normalizedString"/>
    <xs:attribute name="nameStructure" use="optional" default="IEDName">
        <xs:simpleType>
            <xs:restriction base="xs:Name">
                <xs:enumeration value="IEDName"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="tHitem" mixed="true">
    <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" default=""/>
    </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" default=""/>
    </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" default=""/>
    </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>

```

## E.2 Substation syntax

Identical to A.2 (except schema version 2.0)

## E.3 Data type templates

Identical to A.3 (except schema version 2.0)

## E.4 IED capabilities and structure

### SCL\_IED.xsd

This file contains most differences to the 3.0 version. For backward compatibility it allows deprecated options, and sets defaults to instances from 2003 A version.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
    Release 2007/11/19.</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:attributeGroup>

```

```

    <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" fixed="true"/>
    <xs:attribute name="sampleRate" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="dataSet" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="security" 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="entryID" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="configRef" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="bufOvfl" type="xs:boolean" use="optional" default="true"/>
  </xs:attributeGroup>
  <xs:attributeGroup name="agLDRef">
    <xs:attributeGroup ref="scl:agDesc"/>
    <xs:attribute name="iedName" type="tIEDName" use="required"/>
    <xs:attribute name="ldInst" type="tLDInst" use="required"/>
  </xs:attributeGroup>
  <xs:attributeGroup name="agLNRef">
    <xs:attributeGroup ref="agLDRef"/>
    <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
    <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
    <xs:attribute name="lnInst" type="tLNInstOrEmpty" 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="tUnNaming">
        <xs:sequence>
          <xs:element name="Services" type="tServices" minOccurs="0"/>
          <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
            <xs:unique name="uniqueLNlnAccessPoint">
              <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="name" type="tIEDName" use="required"/>
        <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:attribute name="originalSclVersion" type="tSclVersion" use="optional"/>
        <xs:attribute name="originalSclRevision" type="tSclRevision" use="optional"/>
        <xs:attribute name="engRight" type="tRightEnum" use="optional" default="full"/>
        <xs:attribute name="owner" type="xs:normalizedString" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tServices">
    <xs:all>
      <xs:element name="DynAssociation" type="tServiceWithOptionalMax" 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:all>
  </xs:complexType>

```

```

    </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="tServiceForConfDataSet" 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="tServiceConfReportControl" 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="SMV" type="scl:tServiceWithMax" minOccurs="0"/>
  <xs:element name="FileHandling" type="tServiceYesNo" minOccurs="0"/>
  <xs:element name="ConflNs" type="tConflNs" minOccurs="0"/>
  <xs:element name="ClientServices" type="tClientServices" minOccurs="0"/>
  <xs:element name="ConfLdName" type="tServiceYesNo" minOccurs="0"/>
</xs:all>
</xs:complexType>
<xs:complexType name="tAccessPoint">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element name="Server" type="scl:tServer">
            <xs:unique name="uniqueAssociationInServer">
              <xs:selector xpath="/.scl:Association"/>
              <xs:field xpath="@associationID"/>
            </xs:unique>
          </xs:element>
          <xs:element ref="scl:LN" maxOccurs="unbounded"/>
          <xs:element name="ServerAt" type="tServerAt"/>
        </xs:choice>
        <xs:element name="Services" type="scl:tServices" minOccurs="0"/>
        <xs:element name="GOOSESecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
        <xs:element name="SMVSecurity" type="tCertificate" minOccurs="0" maxOccurs="7"/>
      </xs:sequence>
      <xs:attribute name="name" type="tAccessPointName" use="required"/>
      <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="tCertificate">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="Subject" type="tCert"/>
        <xs:element name="IssuerName" type="tCert"/>
      </xs:sequence>
      <xs:attribute name="xferNumber" type="xs:unsignedInt" use="optional"/>
      <xs:attribute name="serialNumber" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:normalizedString">
            <xs:minLength value="1"/>
            <xs:pattern value="\d+"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tCert">
  <xs:attribute name="commonName" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:normalizedString">
        <xs:minLength value="4"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

```

```

        <xs:pattern value="none"/>
        <xs:pattern value="CN=.+"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="idHierarchy" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:normalizedString">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="tServerAt">
    <xs:complexContent>
        <xs:extension base="tUnNaming">
            <xs:attribute name="apName" type="tAccessPointName" use="required"/>
        </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:unique name="uniqueLogName">
                        <xs:selector xpath=".*/*/scl:Log"/>
                        <xs:field xpath="@name"/>
                    </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="tLDInst" use="required"/>
            <xs:attribute name="ldName" type="tLDName" use="optional"/>
        </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:attributeGroup ref="agLNRef"/>
    <xs:attribute name="kind" type="tAssociationKindEnum" use="required"/>
    <xs:attribute name="associationID" type="tName" use="optional"/>
</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:unique name="uniqueDOIinLN0">
        <xs:selector xpath="/scl:DOI"/>
        <xs:field xpath="@name"/>
    </xs:unique>
</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:unique name="uniqueDOIinLN">
        <xs:selector xpath="/scl:DOI"/>
        <xs:field xpath="@name"/>
    </xs:unique>
    <xs:key name="providedLogName">
        <xs:selector xpath="/scl:Log"/>
        <xs:field xpath="@name"/>
    </xs:key>
</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:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

    <xs:element name="DOI" type="tDOI" minOccurs="0" maxOccurs="unbounded">
      <xs:unique name="uniqueSDI_DAlinDOI">
        <xs:selector xpath="/*"/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
    <xs:element name="Inputs" type="tInputs" minOccurs="0">
      <xs:unique name="uniqueExtRefInInputs">
        <xs:selector xpath="./scl:ExtRef"/>
        <xs:field xpath="@iedName"/>
        <xs:field xpath="@ldInst"/>
        <xs:field xpath="@prefix"/>
        <xs:field xpath="@lnClass"/>
        <xs:field xpath="@lnInst"/>
        <xs:field xpath="@doName"/>
        <xs:field xpath="@daName"/>
        <xs:field xpath="@intAddr"/>
      </xs:unique>
    </xs:element>
    <xs:element name="Log" type="scl:tLog" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="lnType" type="tName" use="required"/>
</xs:extension>
</xs:complexType>
</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="tLNInst" use="required"/>
      <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
    </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: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>
<xs:complexType name="tDataSet">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:choice maxOccurs="unbounded">
        <xs:element name="FCDA" type="tFCDA"/>
        <xs:element name="FCCB" type="tFCCB"/>
      </xs:choice>
      <xs:attribute name="name" type="tDataSetName" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tFCDA">
  <xs:attribute name="ldInst" type="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="optional"/>
  <xs:attribute name="lnInst" type="tLNInst" 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:attribute name="ix" type="xs:unsignedInt" use="optional"/>
</xs:complexType>
<xs:complexType name="tFCCB">
  <xs:attribute name="ldInst" type="tLDInst" use="required"/>
  <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
  <xs:attribute name="lnClass" type="tLNClassEnum" use="required"/>
  <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
  <xs:attribute name="cbName" type="tCBName" use="required"/>
  <xs:attribute name="daName" type="tName" use="optional"/>

```

```

    <xs:attribute name="fc" type="tServiceFCEnum" use="required"/>
    <xs:attribute name="mustUnderstand" type="xs:boolean" use="required" fixed="true"/>
  </xs:complexType>
  <xs:complexType name="tControl" abstract="true">
    <xs:complexContent>
      <xs:extension base="tUnNaming">
        <xs:attribute name="name" type="tCBName" use="required"/>
        <xs:attribute name="datSet" type="tDataSetName" use="optional"/>
      </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:attribute name="gi" type="xs:boolean" use="optional" default="true"/>
    <xs:attribute name="includeAll" 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="optional"/>
        <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:attribute name="indexed" type="xs:boolean" use="optional" default="true"/>
      </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="ldInst" type="tLDInst" use="optional"/>
        <xs:attribute name="prefix" type="tPrefix" use="optional" default=""/>
        <xs:attribute name="lnClass" type="scl:LNClassEnum" use="optional" default="LLN0"/>
        <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
        <xs:attribute name="logName" type="tLogName" 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>
<xs:complexType name="tExtRef">
  <xs:attributeGroup ref="scl:agDesc"/>
  <xs:attribute name="iedName" type="tIEDName" use="optional"/>
  <xs:attribute name="ldInst" type="tLDInst" use="optional"/>
  <xs:attribute name="prefix" type="scl:tPrefix" use="optional"/>
  <xs:attribute name="lnClass" type="scl:tLNClassEnum" use="optional"/>
  <xs:attribute name="lnInst" type="tLNInst" use="optional"/>
  <xs:attribute name="doName" type="scl:tName" use="optional"/>
  <xs:attribute name="daName" type="scl: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:attribute name="name" type="tLogName" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tControlWithIEDName">
  <xs:complexContent>
    <xs:extension base="tControl">
      <xs:sequence>
        <xs:element name="IEDName" type="tIEDName" 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:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:unique name="uniqueSDI_DAIinSDI">
            <xs:selector xpath="/*"/>
            <xs:field xpath="@name"/>
        </xs:unique>
    </xs:element>
    <xs:element name="DAI" type="tDAI"/>
</xs:choice>
<xs:attribute name="name" type="tDataName" 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="tAttributeNameEnum" 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="tAttributeNameEnum" 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="tServiceWithOptionalMax">
    <xs:attribute name="max" type="xs:unsignedInt" use="optional"/>
</xs:complexType>
<xs:complexType name="tServiceWithMax">
    <xs:attribute name="max" type="xs:unsignedInt" use="required"/>
</xs:complexType>
<xs:complexType name="tServiceConfReportControl">
    <xs:complexContent>
        <xs:extension base="tServiceWithMax">
            <xs:attribute name="bufMode" use="optional">
                <xs:simpleType>
                    <xs:restriction base="xs:Name">
                        <xs:enumeration value="unbuffered"/>
                        <xs:enumeration value="buffered"/>
                        <xs:enumeration value="both"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="bufConf" type="xs:boolean" use="optional" default="false"/>
        </xs:extension>
    </xs:complexContent>
</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="tServiceForConfDataSet">
    <xs:complexContent>

```

```

    <xs:extension base="tServiceWithMaxAndMaxAttributes">
      <xs:attribute name="modify" type="xs:boolean" use="optional" default="true"/>
      <xs:attribute name="fccb" type="xs:boolean" use="optional" default="false"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tClientServices">
  <xs:attribute name="goose" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="gsse" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="bufReport" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="unbufReport" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="readLog" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="sv" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="fccb" type="xs:boolean" use="optional" default="false"/>
</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="applID" 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:keyref name="ref2LDeviceInDataSetForFCDAinLN" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet/scl:FCDA"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<xs:keyref name="ref2LDeviceInDataSetForFCCBinLN" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet/scl:FCCB"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<xs:keyref name="ref2LDeviceInDataSetForFCDAinLN0" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet/scl:FCDA"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<xs:keyref name="ref2LDeviceInDataSetForFCCBinLN0" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet/scl:FCCB"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<!--<xs:keyref name="ref2LDeviceInDataSetForFCDAinLN" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN/scl:DataSet"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<xs:keyref name="ref2LDeviceInDataSetForFCDAinLN0" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0/scl:DataSet"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref>
<xs:keyref name="ref2LDeviceInDataSetForFCDAinLN2" refer="LDeviceInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:LN/scl:DataSet"/>
  <xs:field xpath="@IdInst"/>
</xs:keyref-->
<xs:key name="AccessPointInIEDKey">
  <xs:selector xpath="/scl:AccessPoint"/>
  <xs:field xpath="@name"/>
</xs:key>
<xs:keyref name="ServerAtRef2AccessPoint" refer="AccessPointInIEDKey">
  <xs:selector xpath="/scl:AccessPoint/scl:ServerAt"/>
  <xs:field xpath="@apName"/>
</xs:keyref>
</xs:element>
</xs:schema>

```

## E.5 Communication subnetworks

Identical to A.5 (except schema version 2.0)

## E.6 Main SCL

### SCL.xsd

Allows all SCL versions. Sets the SCL version / revision default in case of missing attributes to 2003 A, thus being backward compatible.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema 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" targetNamespace="http://www.iec.ch/61850/2003/SCL"
  elementFormDefault="qualified" attributeFormDefault="unqualified" finalDefault="extension" version="3.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Revised SCL normative schema. Version 3.0. (SCL language version "2007").
    Release 2007/11/19.</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:attribute name="version" type="tSclVersion" use="required" fixed="2007"/>
        <xs:attribute name="revision" type="tSclRevision" use="required" fixed="A"/>
      </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:unique name="uniqueLDName">
    <xs:selector xpath="//scl:LDevice"/>
    <xs:field xpath="@ldName"/>
  </xs:unique>
  -->
  <!--
  <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:keyref name="refConnectedAP2IED" refer="IEDKey">
    <xs:selector xpath="/scl:Communication/scl:SubNetwork/scl:ConnectedAP"/>
    <xs:field xpath="@iedName"/>
  </xs:keyref>
</xs:element>
</xs:schema>

```

## Annex F (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, 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.

The SED file is from the syntax restrictions identical to an SCD file.

The IID file for an IED is from the syntax restrictions similar to a CID file, except that typically the concerned IED is the only IED in the file.

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="2.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      COPYRIGHT IEC, 2007. Version 3.0. Release 2007/11/23.
      This schema is for informational purposes only, and is not normative!
      Notes:
      - Identity constraints 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="uniquePowerTransformerInSubstation">
                <xs:selector xpath="./scl:PowerTransformer"/>
              </xs:unique>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    <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="LNNodeTypeKey">
  <xs:selector xpath="./scl:DataTypeTemplates/scl:LNNodeType"/>
  <xs:field xpath="@id"/>
  <xs:field xpath="@lnClass"/>
</xs:key>
<xs:keyref name="ref2LNNodeTypeDomain1" refer="LNNodeTypeKey">
  <xs:selector xpath="./scl:IED/scl:AccessPoint/scl:LN"/>
  <xs:field xpath="@lnType"/>
  <xs:field xpath="@lnClass"/>
</xs:keyref>
<xs:keyref name="ref2LNNodeTypeDomain2" refer="LNNodeTypeKey">
  <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="ref2LNNodeTypeLN0" refer="LNNodeTypeKey">
  <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="ref2LNNodeTypeDomain1" refer="LNNodeTypeKey">
    <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:LN"/>
    <xs:field xpath="@InType"/>
    <xs:field xpath="@InClass"/>
  </xs:keyref>
  <xs:keyref name="ref2LNNodeTypeDomain2" refer="LNNodeTypeKey">
    <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
    <xs:field xpath="@InType"/>
    <xs:field xpath="@InClass"/>
  </xs:keyref>
  <xs:keyref name="ref2LNNodeTypeLLN0" refer="LNNodeTypeKey">
    <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
    <xs:field xpath="@InType"/>
    <xs:field xpath="@InClass"/>
  </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="LNNodeTypeKey">
      <xs:selector xpath="/.scl:DataTypeTemplates/scl:LNNodeType"/>
      <xs:field xpath="@id"/>
      <xs:field xpath="@InClass"/>
    </xs:key>
    <xs:keyref name="ref2LNNodeTypeDomain1" refer="LNNodeTypeKey">
      <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:LN"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </xs:keyref>
    <xs:keyref name="ref2LNNodeTypeDomain2" refer="LNNodeTypeKey">
      <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </xs:keyref>
    <xs:keyref name="ref2LNNodeTypeLLN0" refer="LNNodeTypeKey">
      <xs:selector xpath="/.scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </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="@InClass"/>
    </xs:key>
    <xs:keyref name="ref2LNodeTypeDomain1" refer="LNodeTypeKey">
      <xs:selector xpath="/scl:IED/scl:AccessPoint/scl:LN"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </xs:keyref>
    <xs:keyref name="ref2LNodeTypeDomain2" refer="LNodeTypeKey">
      <xs:selector xpath="/scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </xs:keyref>
    <xs:keyref name="ref2LNodeTypeLLN0" refer="LNodeTypeKey">
      <xs:selector xpath="/scl:IED/scl:AccessPoint/scl:Server/scl:LDevice/scl:LN0"/>
      <xs:field xpath="@InType"/>
      <xs:field xpath="@InClass"/>
    </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:restriction>
  </xs:complexContent>
</xs:complexType>

```

```

        <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:restriction>
        </xs:complexContent>
    </xs:complexType>
</xs:schema>

```

## Annex G (normative)

### SCL Implementation Conformance Statement (SICS)

The following tables contain mandatory and optional features of System Configuration tools and IED configuration tools. It is up to the tool manufacturer to decide to which extent his tool fulfills one or both roles. At least for one main role all mandatory features shall be supported.

The IED configurator features can also partly be implemented within the IED itself, if it can be configured by an SCD or CID file. In this case the conformance statement refers to the combination of IED and IED configurator tool. If an IED tool supports several IED types with different engineering capabilities, then for each combination of tool and IED type a separate IED configurator conformance statement should be given.

The result of an export function can be checked in the generated SCL file. The result of an import can be checked by tool behaviour, and at the final configured IED by browsing through it, or by its communication behaviour.

**Table 50– IED configurator conformance statement**

		Mandatory/ Optional	Value/ comments
<b>ICD export</b>		M	
I11	Fix ICD file (no adaptable export needed)	GC_1 (1)	
I12	Export of ICD file or IID file according to IED preconfiguration performed by tool	GC_1 (1)	
I13	State the data model name space (61850-7-3 clause 7.2) within ICD file (LLN0.NamPIt.IdNs value)	M	
I14	State the data model version (61850-7-3 clause 7.9.3) and any predefined / fixed configuration values within ICD file (9.5.4.4)	M	
I15	Version 2003 export	GC_1 (2)	
I16	Version ____ export	GC_1 (2)	e.g. 2007 for this version
I17	Predefined data sets	O	
I18	Predefined control blocks	O	
I19	Substation bay template with IED part	O	
I110	Communication section with default address	O	
I111	Export correct valKind value (Table 46)	O	RO, Conf
<b>SCD import</b>		M	
I21	Identify IED to be configured in SCD file by IED name	M	
I22	Configure LD name (at least via IdInst, dependent on the IED capabilities) and IED addresses from SCD	M	Support of IdName is stated as IED capability in ICD/IID file; see also I43
I23	Determine communication side addresses of IED inputs from SCD	C1	If data from other IEDs can be received at all
I24	Determine & use clock communication addresses from SCD	C1	If time synch is supported

		Mandatory/ Optional	Value/ comments
I25	Configure values of (existing) control block from SCD(9.3)	C3	
I26	Prepare (new) control block instances according to SCD file	C3	
I27	Prepare / configure data sets according to SCD file	C3	
I28	Modify predefined data sets according to SCD	C3	
I29	Interpret client references in the control blocks of other IEDs to find the control block instances allocated to this IED, and data sent to this IED.	C1	If data from other IEDs can be received at all
I210	Set IED configuration values and parameter values as defined in SCD file	O	
I211	Support changed (reduced capability) valKind (e.g. from <i>Set</i> to <i>RO</i> or to <i>Conf</i> ) (Table 46)	O	
I212	Support IdName on other IEDs (9.3.4)	C3	
<b>IID export after IED engineering</b>		O	
I31	IED version & instance information: LPHD.PhyNam: hwRev, swRev, serNum, LLN0.NamPit.configRev	O	
I32	Configuration values (fc=CF)	O	
I33	Setting Parameter values (fc=SP, SG)	O	
I34	SCL Header management (9.1)	C2	
I35	Modify IED data model (add LN/DATA/LD, or remove unused LD/LN/DATA)	O	Removal of used (referenced) DATA / CB instances is not allowed
<b>Tool functionality</b>		M	
I41	Support MustUnderstand concept (8.2)	M	
I42	Bind incoming 61850 signals to IED internal (input) signals	C1	
I43	Support FCCB elements (9.3.7) in own data sets	C3	
I44	Support FCCB elements in data sets of other IEDs	C3	
I45	Use or create IED Input section for binding incoming (external) signals to internal signals, to document this binding.	O	
I46	Create CID file for IED	O	
I47	Support IdName for LD name specification	C3	
I48	Modify LN prefixes or IdInst	O	Specify what
<p>C1 – mandatory, if the IED can receive data from other IEDs, i.e. be either client or subscriber</p> <p>C2 – mandatory, if any of the other features in this table section is supported</p> <p>C3 – mandatory, if the appropriate IED capability is claimed in PIXIT or IED capability section</p> <p>GC_1 (n)– at least one of the elements of group n shall be available.</p> <p>O – Optional; should match the IED capabilities; i.e. if an IED claims that RCBs can be configured by SCL, then the IED tool shall support it.</p> <p>M – Mandatory</p>			

**Table 51 – System configurator conformance statement**

		Mandatory/ Optional	Value/ comments
<b>ICD&amp;IID import &amp; usage</b>		M	
S11	IED data model	M	
S12	Predefined data sets	M	
S13	Predefined control blocks	M	
S14	Support MustUnderstand concept (8.2)	M	
S15	Support SCL version ____ as input	C1	Version 2003 input is always mandatory
S16	Substation bay template with LN links, if it exists	O	
S17	Reuse already imported DataTypeTemplates for identical types	O	
S18	Keep attributes and elements of unknown XML name spaces outside Private elements for SCD export	O	
S19	Import single line layout coordinates defined in C.1	O	For own usage, or just later export
S111	Import of IID file; update configuration values & setting values, modify data model (see 10.1) .	O	
<b>Communication engineering</b>		M	
S21	Configure (edit) IED names	M	
S22	Create and configure Subnetworks and IED communication addresses	M	
S23	Create / import client IEDs, master clocks, switches and routers	M	
S24	Create and configure physical connection attributes (9.4.6)	O	
S25	Configure IdName values	C1	
S26	Configure IdInst and LN prefixes, if the IED allows this	O	
<b>Data flow engineering</b>		M	
S31	Configure control blocks	M	
S32	Create control block types / instances, if IED capabilities allow so	M	
S33	Create data sets, if IED capability allows this	M	
S34	Modify predefined or created data sets	M	
S35	Manage control block confRev	M	
S36	Allocate control block instances to clients & define data destinations ( <i>ClientLn</i> element, <i>IEDName</i> element)	M	
S37	Edit Input sections in LNs	O	
S38	Create Input section from configured data flow	O	

		Mandatory/ Optional	Value/ comments
<b>SCD Substation section handling</b>		O	
S41	Import Substation section from SSD/SCD file	O	
S42	Edit / create substation section	O	
S43	Bind logical nodes from IEDs to substation section	O	
S44	Create bay instances from IED / ICD substation templates	O	
S45	Edit / create substation topology (connections between primary equipment (9.2.4))	O	
S46	Edit substation element names and desc attributes.	O	
S47	Edit Equipment terminals ( <i>name</i> attribute of the <i>Terminal</i> element)	O	
S48	Edit / create the <i>Function</i> / <i>SubFunction</i> / <i>GeneralEquipment</i> naming hierarchies	O	
<b>SCD Modifications</b>		M	
S51	Handle SCD Header revision history; mark change	M	
S52	Set configuration values (attributes with fc=CF, DC)	O	
S53	Set Parameter (Setting DATA) values, also for different setting groups	O	
S54	Add / Modify layout coordinates according to C.1	O	Substation section, Communication section
S55	Show IED service / engineering capabilities to engineer	O	
S56	Interprete IED capabilities and prohibit unsupported usage	O	
S57	Editing of data attributes valKind property (9.5.4.1)	O	
<b>SCD export</b>		M	
S61	Version 2003 export	GC_1(1)	
S62	Version 2007 export	GC_1(1)	
S63	Version _____ export	O	Future versions
S64	Restore imported Private sections	M	
S65	Export restructured DataTypeTemplate types (keeping all instance related information and all Private information constant)	O	
<b>SCD Import</b>		O	
S71	Version 2003 import	GC_1(2)	
S72	Version 2007 import	GC_1(2)	
S73	Version _____ import	O	Future versions
S74	Add new bays / equipment in substation section	O	
S75	Add links from substation section to IEDs	O	
S77	Update IED configuration values	O	
S78	Update IED Setting DATA values	O	

		Mandatory/ Optional	Value/ comments
S79	Add new IEDs	O	
<b>SED handling (5.5, 10.3)</b>		O	
S81	Export SED for selected dataflow IEDs	M	
S82	Import SED with exported part	M	
S83	Prohibit editing of IEDs exported with dataflow right (inclusive Substation section links & communication addresses)	M	
S84	Import SED for usage in own project; export modified SED back to source project	O	
S85	Import / merge Substation section part from SED	O	
S86	Import / merge Communication section part from SED	O	
<p>C1 – mandatory for tools supporting the 2007 SCL version</p> <p>GC_1(n) – at least one of the elements of group (n) should be available.</p> <p>O – Optional; should match the IED capabilities; i.e. if an IED claims that RCBs can not be configured by SCL, then the tool shall prohibit it.</p> <p>M – Mandatory; should match the IED capabilities; i.e. if an IED claims that RCBs can not be configured by SCL, then the tool shall prohibit it.</p>			

## **Bibliography**

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

---