

DIN EN 61850-6



ICS 33.040.40

Ersatz für
DIN EN 61850-6:2010-06
Siehe Anwendungsbeginn

**Kommunikationsnetze und -systeme für die Automatisierung in der elektrischen Energieversorgung –
Teil 6: Sprache für die Beschreibung der Konfiguration für die Kommunikation in Stationen mit intelligenten elektronischen Geräten (IED)
(IEC 61850-6:2009 + A1:2018);
Englische Fassung EN 61850-6:2010 + A1:2018**

Communication networks and systems for power utility automation –
Part 6: Configuration description language for communication in power utility automation systems related to IEDs
(IEC 61850-6:2009 + A1:2018);
English version EN 61850-6:2010 + A1:2018

Systèmes et réseaux de communication pour l'automatisation des services de distribution d'énergie –
Partie 6: Langage pour la description de configuration pour la communication dans les postes électriques, entre les dispositifs électroniques intelligents (IED)
(IEC 61850-6:2009 + A1:2018);
Version anglaise EN 61850-6:2010 + A1:2018

Gesamtumfang 275 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2010-02-01 angenommene Europäische Norm und die am 2018-07-12 angenommene Änderung A1 als DIN-Norm ist 2020-03-01.

Für DIN EN 61850-6:2010-06 besteht eine Übergangsfrist bis 2021-07-12.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61850-6/A1:2018-05.

Für dieses Dokument ist das nationale Arbeitsgremium K 952 „Netzleittechnik“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom TC 57 „Power systems management and associated information exchange“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieses Dokuments bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu diesem Dokument angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees das Dokument

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Die Änderung A1 wurde eingearbeitet und durch einen senkrechten Strich am linken Seitenrand gekennzeichnet.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61850-6 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

Änderungen

Gegenüber DIN EN 61850-6:2010-06 wurden folgende Änderungen vorgenommen:

- a) Zusammenfassend zielen die Änderungen darauf ab, den ursprünglichen Inhalt der IEC 61850-6:2009 zu ändern und seinen Inhalt zu korrigieren, um eine bessere Interoperabilität der Komponenten sicherzustellen, die ihn umsetzen. Es handelt sich um zwei verschiedene Arten von Änderungen:
- b) gelöste Interop-Probleme (die unter Klausel 3 dieses Änderungsantrags 1 fallen), die bereits den in IEC 61850-1 beschriebenen Prozess der technischen Probleme (Gewebe) verfolgt haben und den grünen „Status“ erreicht haben;
- c) gelöste redaktionelle Angelegenheiten (technische Fragen), die zu Interoperabilitätsproblemen geführt haben könnten.

Frühere Ausgaben

DIN EN 61850-6: 2005-12, 2010-06

Nationaler Anhang NA (informativ)

Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

1 Anwendungsbereich

Der vorhandene Text des Anwendungsbereichs wird durch den folgenden neuen Text ersetzt:

1.1 Allgemeines

Dieser Teil der IEC 61850 legt ein Dateiformat zur Beschreibung kommunikationsbezogener IED-Konfigurationen (en: Intelligent Electronic Device, de: intelligentes elektronisches Gerät) und IED-Parameter, Konfigurationen von Kommunikationssystemen, Strukturen von Schaltanlagen(-funktionen) und deren Beziehungen untereinander fest. Dieses Format soll hauptsächlich dem Zweck dienen, IED-Funktionsbeschreibungen (en: IED Capability Description) und SA-Systembeschreibungen zwischen IED-Engineeringtools und den System-Engineeringtools verschiedener Hersteller kompatibel auszutauschen.

Die definierte Sprache wird als „System Configuration Description Language“ (SCL, de: Sprache zur Beschreibung von Systemkonfigurationen) bezeichnet. Das IED- und Kommunikationssystemmodell in SCL entspricht IEC 61850-5 und IEC 61850-7-x. In den entsprechenden Teilen können SCSSM-spezifische Erweiterungen oder Anwendungsregeln erforderlich sein.

Die Konfigurationssprache beruht auf XML (en: Extensible Markup Language), Version 1.0 (siehe XML-Verweisungen in Abschnitt 2).

Diese Norm legt weder einzelne Implementierungen oder Produkte fest, die diese Sprache verwenden, noch enthält sie Beschränkungen hinsichtlich der Implementierung von Einheiten und Schnittstellen innerhalb eines Computersystems. Dieser Teil der Norm legt nicht das Format zum Herunterladen von Konfigurationsdaten auf ein IED fest, obwohl er für einen Teil der Konfigurationsdaten verwendet werden könnte.

1.2 Name und Version des Namensraums

Dieser neue Abschnitt ist für alle Namensräume nach IEC 61850 obligatorisch (wie in IEC 61850-7-1:2011 festgelegt).

Die Parameter zur Identifikation dieser neuen Freigabe des SCL-Namensraums

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

- Version des Namensraums: 2007
- Überarbeitung des Namensraums: B
- Freigabe des Namensraums: 4
- Freigabedatum des Namensraums: 2018/01/22

Die unten stehende Tabelle enthält eine Übersicht über alle veröffentlichten Versionen dieses Namensraums.

Ausgabe	Veröffentlichungsdatum	Webstore	Namensraum
Edition 1.0	2004-03	IEC 61850-6:2004	IEC 61850-6:2003
Edition 2.0	2009-12	IEC 61850-6:2009	IEC 61850-6:2007B
Amendment 1 von Edition 2.0	2018	IEC 61850-6:2009/AMD1:2018	IEC 61850-6:2007B4
Edition 2.1	2018	IEC 61850-6:2009+AMD1:2018 CSV	IEC 61850-6:2007B4

1.3 Verbreitung der Code-Komponenten

Die in dieser IEC-Norm enthaltenen Code-Komponenten sind auch als maschinenlesbare Datei unter folgender Adresse erhältlich:

http://www.iec.ch/tc57/supportdocuments/IEC_61850-6.2018.SCL.2007B4.full.zip

Die in dieser IEC-Norm enthaltene(n) Code-Komponente(n) wird (werden) möglicherweise derzeit überarbeitet, sodass Anwender die neueste Freigabe in der Datenbank unter folgender Adresse auswählen müssen:

<http://www.iec.ch/tc57/supportdocuments>

Die neueste Version/Freigabe des Dokuments wird durch Auswahl der Datei „IEC_61850-6.2018.SCL.{VersionStateInfo}.full.zip“ gefunden, wobei das Feld „VersionStateInfo“ den höchsten Wert enthält.

Jede Code-Komponente ist als ZIP-Paket verfügbar und enthält die elektronische Darstellung der Code-Komponente selbst sowie eine Datei, die den Inhalt des Pakets beschreibt („IECManifest.xml“).

„IECManifest.xml“ enthält verschiedene Abschnitte mit Informationen zu folgenden Punkten:

- Urheberrechtsvermerk;
- Identifikation der Code-Komponente;
- Veröffentlichung im Zusammenhang mit der Code-Komponente;
- Liste der elektronischen Dateien, die die Code-Komponente bilden;
- optionale Liste der historischen Dateien zur Nachverfolgung der Änderungen während des Entwicklungsprozesses der Code-Komponente.

Die zu dieser Veröffentlichung gehörende „IECManifest“-Datei ist folgende:

The IECManifest related to this publication is:

```
<IECManifest xmlns="http://www.iec.ch/CC/2017/IECManifest" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/CC/2017/IECManifest IECManifest.xsd">
  <Copyright>
    <Notice>
      COPYRIGHT (c) IEC, 2017. This version of this XSD is part of IEC 61850-6:2009/AMD1:2018; see the IEC 61850-
      6:2009/AMD1:2018 for full legal notices. In case of any differences between the here-below code and the IEC published content, the
      here-below definition supersedes the IEC publication; it may contain updates. See history files. The whole document has to be taken into
      account to have a full description of this code component.
      See www.iec.ch/CCv1 for copyright details.
    </Notice>
    <License uri="www.iec.ch/CCv1">IEC License</License>
  </Copyright>
  <CodeComponent id="IEC_61850-6.2007B4.SCL.XSD" name="IEC 61850-6 SCL schema V2007B4" content="full" date="2018-01-
  22">
    <Publication name="IEC_61850-6.2018_ed2.1" comment="Configuration description language for communication in power
    utility automation systems related to IEDs"/>
    <File name="SCL.xsd" category="normative" content="full"/>
    <File name="SCL_Substation.xsd" category="normative" content="full"/>
    <File name="SCL_Communication.xsd" category="normative" content="full"/>
    <File name="SCL_IED.xsd" category="normative" content="full"/>
    <File name="SCL_DataTypeTemplates.xsd" category="normative" content="full"/>
    <File name="SCL_BaseTypes.xsd" category="normative" content="full"/>
    <File name="SCL_BaseSimpleTypes.xsd" category="normative" content="full"/>
    <File name="SCL_Enums.xsd" category="normative" content="full"/>
    <File name="SCL.Doc.HTML.zip" category="normative" content="full" comment="Zip archive containing the HTML
    documentation of the SCL. Contains the 'SCL.html' file and all related pictures"/>
    <HistoryFile name="history.2007B4.txt" startingDate="2014-06-11" endingDate="2017-07-05"
    startingVersion="SCL.XSD.v2007B" endingVersion="SCL.XSD.v2007B4"/>
  </CodeComponent>
</IECManifest>
```

Das Paket wird mithilfe der folgenden Namensregel identifiziert:

{RefStandard}.{CodeComponentName}.{VersionRevision}.{LightFull}{PublicationStage}.zip Für die aktuelle Veröffentlichung lautet der Name des Code-Komponenten-Pakets:

IEC_61850-6.2018.SCL.2007B4.full.zip

Der Lebenszyklus einer Code-Komponente hängt nicht vom Lebenszyklus der zugehörigen Veröffentlichung ab. Der Lebenszyklus der Veröffentlichung umfasst zwei Stufen: Version (entsprechend einer Ausgabe) und Überarbeitung (entsprechend einer Änderung). Mithilfe einer dritten Veröffentlichungsstufe (Freigabe) kann eine Code-Komponente veröffentlicht werden, ohne dass eine Änderung herausgegeben werden muss.

Das ist hilfreich, wenn die Interoperabilität betreffende Fehler („InterOp Tissues“) behoben werden müssen. Dann wird eine neue Freigabe der Code-Komponente veröffentlicht, welche die vorhergehende Freigabe ersetzt, und über die Website des IEC TC 57 verbreitet.

English Version

Communication networks and systems for power utility
automation - Part 6: Configuration description language for
communication in power utility automation systems related to
IEDs
(IEC 61850-6:2009 + A1:2018)

Systèmes et réseaux de communication pour
l'automatisation des services de distribution d'énergie -
Partie 6: Langage pour la description de configuration pour
la communication dans les postes électriques, entre les
dispositifs électroniques intelligents (IED)
(IEC 61850-6:2009 + A1:2018)

Kommunikationsnetze und -systeme für die
Automatisierung in der elektrischen Energieversorgung -
Teil 6: Sprache für die Beschreibung der Konfiguration für
die Kommunikation in Stationen mit intelligenten
elektronischen Geräten (IED)
(IEC 61850-6:2009 + A1:2018)

This European Standard was approved by CENELEC on 2010-02-01 and A1 on 2018-07-12. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this amendment the status of a national standard without any alteration.

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

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

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



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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Foreword

The text of document 57/1025/FDIS, future edition 2 of IEC 61850-6, prepared by IEC TC 57, Power systems management and associated information exchange, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61850-6 on 2010-02-01.

This European Standard supersedes EN 61850-6:2004.

The main changes with respect to EN 61850-6:2004 are as follows:

- functional extensions added based on changes in other Parts, especially Parts 7-2 and 7-3;
- functional extensions concerning the engineering process, especially for configuration data exchange between system configuration tools, added;
- provision of clarifications and corrections. Issues that require clarification are published in a database available at www.tissue.iec61850.com. Arising incompatibilities are listed in 8.2.3.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN and CENELEC shall not be held responsible for identifying any or all such patent rights.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2010-11-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2013-02-01

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 61850-6:2009 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

- | | |
|-------------|--------------------------------|
| IEC 61131-3 | NOTE Harmonized as EN 61131-3. |
| IEC 81346-2 | NOTE Harmonized as EN 81346-2. |

European foreword A1

The text of document 57/1918/FDIS, future edition 2 of IEC 61850-6/A1, prepared by IEC/TC 57 "Power systems management and associated information exchange" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61850-6:2010/A1:2018.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2019-04-12
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2021-07-12

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61850-6:2009/A1:2018 was approved by CENELEC as a European Standard without any modification.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC/TS 61850-2	–	Communication networks and systems in substations – Part 2: Glossary	–	–
IEC 61850-4	–	Communication networks and systems for power utility automation – Part 4: System and project management	EN 61850-4	–
IEC 61850-5	–	Communication networks and systems in substations – Part 5: Communication requirements for functions and device models	EN 61850-5	–
IEC 61850-7-1	2011	Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models	EN 61850-7-1	2011
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)	EN 61850-7-2	–
IEC 61850-7-3	–	Communication networks and systems in substations – Part 7-3: Basic communication structure for substation and feeder equipment – Common data classes	EN 61850-7-3	–
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	EN 61850-7-4	–
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	EN 61850-8-1	–
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	EN 61850-9-2	–
IEC/IEEE 61850-9-3	–	Communication networks and systems for power utility automation – Part 9-3: Precision time protocol profile for power utility automation	–	–
IEC 81346-1	–	Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules	EN 81346-1	–

ISO/IEC 8859-1	–	Information technology – 8-bit single-byte coded – graphic character sets – Part 1: Latin alphabet No.1	–	–
RFC 1952	–	GZIP file format specification version 4.3	–	–
RFC 2045	–	Multipurpose Internet Mail Extensions (MIME) Part 1: Format of Internet Message Bodies	–	–
–	–	Extensible Markup Language (XML) 1.0	–	–
–	–	XML Schema Part 1: Structures	–	–
–	–	XML Schema Part 2: Datatypes	–	–

CONTENTS

Foreword.....	2
European foreword A1.....	3
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	4
INTRODUCTION	12
1 Scope.....	13
1.1 General	13
1.2 Namespace name and version	13
1.3 Code Component distribution	13
2 Normative references	15
3 Terms and definitions	16
4 Abbreviations	17
5 Intended engineering process with SCL.....	17
5.1 General	17
5.2 Scope of SCL.....	18
5.3 Use of SCL in the engineering process	18
5.4 IED modifications	22
5.5 Data exchange between projects	23
6 The SCL object model	25
6.1 General	25
6.2 The process model	29
6.3 The product (IED) model	30
6.4 The communication system model.....	31
6.5 Modelling of redundancy.....	32
6.6 Data flow modelling.....	32
7 SCL description file types	33
8 SCL language	35
8.1 Specification method.....	35
8.2 Language versions and compatibility.....	37
8.2.1 MustUnderstand rules	38
8.2.2 SCL name space and versions.....	39
8.2.3 Incompatibilities to earlier versions.....	40
8.3 SCL language extensions	41
8.3.1 General	41
8.3.2 Data model extensions	41
8.3.3 Additional semantics to existing syntax elements	41
8.3.4 Data type constraints.....	41
8.3.5 XML name spaces	41
8.3.6 Private data.....	42
8.3.7 Another XML syntax	43
8.3.8 Summary: Standard conformance for extension handling	43
8.3.9 Extension example	43
8.4 General structure	43
8.5 Object and signal designation.....	44

8.5.1	General.....	44
8.5.2	Object designations in an object hierarchy	44
8.5.3	Signal identifications to be used in the communication system	45
8.5.4	Signal identifications usable by applications	47
8.5.5	Naming example	47
9	The SCL syntax elements	48
9.1	Header	48
9.2	Process description	51
9.2.1	General.....	51
9.2.2	Voltage level.....	56
9.2.3	Bay level	57
9.2.4	Power equipment	58
9.2.5	SubEquipment level	65
9.2.6	Process function logical nodes.....	66
9.2.7	Non power equipment	67
9.2.8	Substation section example	68
9.3	IED description	70
9.3.1	General.....	70
9.3.2	The IED, Services and Access Point	74
9.3.3	The IED server	86
9.3.4	The logical device.....	87
9.3.5	LN0 and other Logical Nodes.....	88
9.3.6	Data object (DOI) definition	90
9.3.7	Data set definition.....	93
9.3.8	Report control block	95
9.3.9	Log control block	98
9.3.10	GSE control block.....	99
9.3.11	Sampled value control block	101
9.3.12	Setting control block	103
9.3.13	Binding to external signals	104
9.3.14	Associations	109
9.4	Communication system description.....	109
9.4.1	General.....	109
9.4.2	Subnetwork definition	110
9.4.3	Address definition	112
9.4.4	GSE address definition.....	113
9.4.5	SMV address definition	114
9.4.6	Physical connection parameters	114
9.4.7	Communication section example	115
9.5	Data type templates.....	116
9.5.1	General.....	116
9.5.2	LNodeType definitions.....	119
9.5.3	DO type definition	120
9.5.4	Data attribute (DA) definition	121
9.5.5	Data attribute structure type	126
9.5.6	Enumeration types	127
9.5.7	Data type template examples.....	128

10	Tool and project engineering rights	128
10.1	IED configurator	129
10.2	System configurator	129
10.3	Right transfer between projects	130
Annex A (normative) SCL syntax: XML schema definition		132
Annex B (informative) SCL enumerations according to IEC 61850-7-3 and IEC 61850-7-4		185
Annex C (informative) Syntax extension examples		186
C.1	Extension syntax for drawing layout coordinates	186
Annex D (informative) Example		188
D.1	Example specification	188
D.1.1	General	188
D.1.2	Substation configuration	188
D.1.3	Communication system configuration	189
D.1.4	Transformer IED	189
D.2	Example SCL file contents	190
Annex E (informative) SCL syntax: General XML schema definition		203
E.1	General	203
E.2	Base types	203
E.3	Substation syntax	219
E.4	Data type templates	219
E.5	IED capabilities and structure	219
E.6	Communication subnetworks	232
E.7	Main SCL	232
Annex F (informative) XML schema definition of SCL variants		234
Annex G (normative) SCL Implementation Conformance Statement (SICS)		240
Annex H (informative) ExtRef use cases		245
Annex I (normative) SCL – mixed version projects		249
I.1	General	249
I.2	Downgrading rules	252
I.2.1	Data Model related downgrading rules	252
I.2.2	SCL IED and Substation Model related compatibility rules	256
I.2.3	Generic rules	258
I.3	Upgrading rules	259
I.3.1	Data Model Structure related upgrading rules	259
I.3.2	SCL Element related upgrading rules	260
I.4	Mixed version projects involving Ed2, Ed2.1 devices / SCTs	262
I.4.1	General	262
I.4.2	Downgrading rules	265
I.4.3	Upgrading rules	267
Bibliography		269
Figure 1 – Reference model for information flow in the configuration process		20
Figure 2 – IED type description to System Configurator		21
Figure 3 – IED instance description to System Configurator		22
Figure 4 – Modification process		23

Figure 5 – Engineering right handling in projects	24
Figure 6 – SCL substation object model	26
Figure 7 – SA System Configuration example	29
Figure 8 – ICD files describing implementable IED types of a general IED class.....	35
Figure 9 – UML diagram overview of SCL schema.....	37
Figure 10 – Elements of the signal identification as defined in IEC 61850-7-2.....	45
Figure 11 – Elements of the signal name using product naming	46
Figure 12 – Possible elements of the signal name using functional naming	46
Figure 13 – Names within different structures of the object model	48
Figure 14 – UML diagram of Header section	49
Figure 15 – UML diagram of Substation section	52
Figure 16 – UML diagrams for equipment type inheritance and relations	59
Figure 17 – Substation section example	69
Figure 18 – IED structure and access points	71
Figure 19 – UML description of IED-related schema part – Base.....	72
Figure 20 – UML description of IED-related schema part for Control blocks.....	73
Figure 21 – UML description of IED-related schema part – LN definition.....	74
Figure 22 – UML diagram overview of the Communication section.....	110
Figure 23 – UML overview of DataTypeTemplate section	117
Figure 24 – UML diagram of Process and Line elements	55
Figure C.1 – Coordinate example	186
Figure D.1 – T1-1 Substation configuration.....	188
Figure D.2 – T1-1 Communication configuration.....	189
Figure D.3 – T1-1 Transformer bay	190
Figure I.1 – Edition 1-Edition 2 – Area of compatibility	249
Figure I.2 – Edition 1-Edition 2 Mixed engineering with different SCL versions	250
Figure I.3 – Edition 1-Edition 2 Mixed engineering with different SCL versions with one SCT managing data flow restriction	251
Figure I.4 – Edition 1-Edition 2 Mixed engineering with same SCL version – restricted to (Ed1∩Ed2)UEd2.....	252
Figure I.5 – Workflow with SED 2007B4 import following mustUnderstand/mayIgnore rules	263
Figure I.6 – Workflow with ICT A imports SCD 2007B4 following the mustUnderstand rules	264
Figure I.7 – Workflow with SCT exports SCD 2007B following the downgrading rules	265
Table 1 – The files composing the XML schema definition for SCL	36
Table 2 – Attributes of the Private element	42
Table 3 – Attributes of the Header element.....	50
Table 4 – Attributes of the History item (Hitem) element	51
Table 5 – Primary apparatus device type codes	63
Table 6 – Attributes of the Terminal element	65
Table 7 – Attributes of the SubEquipment element.....	66
Table 8 – Attributes of the LNode element.....	67
Table 9 – General Equipment codes from IEC 61850-7-4.....	68

Table 10 – Attributes of the IED element.....	75
Table 11 – List of service capabilities and setting elements and attributes	77
Table 12 – Attributes of the Access point element	83
Table 13 – Attributes of the IED server element.....	87
Table 14 – Attributes of the Authentication element.....	87
Table 15 – Attributes of the LDevice element.....	88
Table 16 – Attributes of the LN0 element	89
Table 17 – Attributes of the LN element	90
Table 18 – Attributes of the DOI element	91
Table 19 – Attributes of the DAI element.....	92
Table 20 – Attributes of the SDI element.....	92
Table 21 – Attributes of the DataSet element.....	93
Table 22 – Attributes of the FCDA element.....	94
Table 23 – Attributes of the report control block element.....	95
Table 24 – Attributes of the RptEnabled element.....	97
Table 25 – Attributes of the ClientLN element.....	97
Table 26 – Attributes of the log control block element.....	99
Table 27 – Attributes of the GSE control block element.....	100
Table 28 – Attributes of the IEDName element	101
Table 29 – Attributes of the sampled value control block element.....	102
Table 30 – Attributes of the Smv Options element	103
Table 31 – Deprecated Smv options	103
Table 32 – Attributes of the setting control block element.....	104
Table 33 – Attributes of the Input/ExtRef element.....	106
Table 34 – Attributes of the association element.....	109
Table 35 – Attributes of the Subnetwork element.....	111
Table 36 – Attributes of the ConnectedAP element	112
Table 37 – Attributes of the GSE element	113
Table 38 – Attributes of the SMV element.....	114
Table 39 – PhysConn P-Type definitions	115
Table 40 – Template definition elements.....	119
Table 41 – Attributes of the LNodeType element.....	119
Table 42 – Attributes of the DO element	120
Table 43 – Attributes of the DOType element	120
Table 44 – Attributes of the SDO element.....	121
Table 45 – Data type mapping.....	122
Table 46 – Attribute value kind (Valkind) meaning	123
Table 47 – Attributes of the DA element.....	124
Table 48 – Attributes of the BDA element	127
Table 49 – Attributes of the EnumType element	128
Table 50 – Usage of Service element at IED level and Server / ServerAt level.....	83
Table 51 – Usage of ExtRef attributes in different use cases	108

Table 52 – Allowed SCT engineering actions 131

Table G.1 – IED configurator conformance statement..... 240

Table G.2 – System configurator conformance statement..... 242

INTRODUCTION

This part of IEC 61850 specifies a description language for the configuration of power utility IEDs. This language is called System 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 utility automation system and the process (substation, switch yard). At the application level, the switch yard topology itself and the relation of the switch yard structure to the SAS functions (logical nodes) configured on the IEDs can be described.

While this part describes the language to describe the configuration of IEC 61850 systems, other parts of the standard describe how to configure the system and possible restrictions. Therefore implementations claiming conformance to this standard shall take into account constraints from the other normative references. Some references to the other parts have been included for the purpose of clarification but these references are not all inclusive.

NOTE The process description, which is in this standard 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).

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 simply by restrictions on the way the values of objects have to be used.

1 Scope

1.1 General

This part of IEC 61850 specifies a file format for describing communication-related IED (Intelligent Electronic Device) configurations and IED parameters, communication system configurations, switch yard (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 System 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 (see XML references in Clause 2).

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.

1.2 Namespace name and version

This new section is mandatory for any IEC 61850 namespace (as defined by IEC 61850-7-1:2011).

The parameters which are identifying this new release of the SCL namespace `xmlns:scl="http://www.iec.ch/61850/2003/SCL"` are:

- Namespace Version: 2007
- Namespace Revision: B
- Namespace Release: 4
- Namespace release date: 2018/01/22

The table below provides an overview of all published versions of this namespace.

Edition	Publication date	Webstore	Namespace
Edition 1.0	2004-03	IEC 61850-6:2004	IEC 61850-6:2003
Edition 2.0	2009-12	IEC 61850-6:2009	IEC 61850-6:2007B
Amendment 1 of Edition 2.0	2018	IEC 61850-6:2009/AMD1:2018	IEC 61850-6:2007B4
Edition 2.1	2018	IEC 61850-6:2009+AMD1:2018 CSV	IEC 61850-6:2007B4

1.3 Code Component distribution

The Code Components included in this IEC standard are also available as electronic machine readable file at:

http://www.iec.ch/tc57/supportdocuments/IEC_61850-6.2018.SCL.2007B4.full.zip

The Code Component(s) included in this IEC standard are potentially subject to maintenance works and user shall select the latest release in the repository located at:
<http://www.iec.ch/tc57/supportdocuments>

The latest version/release of the document will be found by selecting the file IEC_61850-6.2018.SCL.{VersionStateInfo}.full.zip with the filed VersionStateInfo of the highest value.

Each Code Component is a ZIP package containing the electronic representation of the Code Component itself, with a file describing the content of the package (IECManifest.xml).

The IECManifest contains different sections giving information on:

- The copyright notice
- The identification of the code component
- The publication related to the code component
- The list of the electronic files which compose the code component
- An optional list of history files to track changes during the evolution process of the code component

The IECManifest related to this publication is:

```
<IECManifest xmlns="http://www.iec.ch/CC/2017/IECManifest" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/CC/2017/IECManifest IECManifest.xsd">
  <Copyright>
    <Notice>
      COPYRIGHT (c) IEC, 2017. This version of this XSD is part of IEC 61850-6:2009/AMD1:2018; see the
      IEC 61850-6:2009/AMD1:2018 for full legal notices. In case of any differences between the here-below code and the IEC published
      content, the here-below definition supersedes the IEC publication; it may contain updates. See history files. The whole document has to
      be taken into account to have a full description of this code component.
      See www.iec.ch/CCv1 for copyright details.
    </Notice>
    <License uri="www.iec.ch/CCv1">IEC License</License>
  </Copyright>
  <CodeComponent id="IEC_61850-6.2007B4.SCL.XSD" name="IEC 61850-6 SCL schema V2007B4" content="full"
date="2018-01-22">
    <Publication name="IEC_61850-6.2018_ed2.1" comment="Configuration description language for communication in
power utility automation systems related to IEDs"/>
    <File name="SCL.xsd" category="normative" content="full"/>
    <File name="SCL_Substation.xsd" category="normative" content="full"/>
    <File name="SCL_Communication.xsd" category="normative" content="full"/>
    <File name="SCL_IED.xsd" category="normative" content="full"/>
    <File name="SCL_DataTypeTemplates.xsd" category="normative" content="full"/>
    <File name="SCL_BaseTypes.xsd" category="normative" content="full"/>
    <File name="SCL_BaseSimpleTypes.xsd" category="normative" content="full"/>
    <File name="SCL_Enums.xsd" category="normative" content="full"/>
    <File name="SCL.Doc.HTML.zip" category="normative" content="full" comment="Zip archive containing the HTML
documentation of the SCL. Contains the 'SCL.html' file and all related pictures"/>
    <HistoryFile name="history.2007B4.txt" startDate="2014-06-11" endingDate="2017-07-05"
startingVersion="SCL.XSD.v2007B" endingVersion="SCL.XSD.v2007B4"/>
  </CodeComponent>
</IECManifest>
```

The package is identified using the following naming rule:

{RefStandard}.{CodeComponentName}.{VersionRevision}.{LightFull}{PublicationStage}.zip For current publication, the Code Component package name is:
IEC_61850-6.2018.SCL.2007B4.full.zip

The life cycle of a code component is not restricted to the life cycle of the related publication. The publication life cycle goes through two stages, Version (corresponding to an edition) and Revision (corresponding to an amendment). A third publication stage (Release) allow publication of Code Component without need to publish an amendment.

This is useful when InterOp Tissues need to be fixed. Then a new release of the Code Component will be released, which supersedes the previous release, and distributed through the IEC TC57 web site.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 TS 61850-2, *Communication networks and systems in substations – Part 2: Glossary*

IEC 61850-4, *Communication networks and systems for power utility automation – Part 4: System and project management*

IEC 61850-5, *Communication networks and systems for power utility automation – Part 5: Communication requirements for functions and device models*

IEC 61850-7-1:2011, *Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models*

IEC 61850-7-2, *Communication networks and systems for power utility automation – Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSI)*

IEC 61850-7-3, *Communication networks and systems for power utility automation – Part 7-3: Basic communication structure – Common data classes*

IEC 61850-7-4, *Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes*

IEC 61850-8-1, *Communication networks and systems for power utility automation – 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 61850-9-2, *Communication networks and systems for power utility automation – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3*

IEC IEEE 61850-9-3, *Communication networks and systems for power utility automation – Part 9-3: Precision time protocol profile for power utility automation*

IEC 81346-1, *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules*

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

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>>

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

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/>>

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. Only general meanings are given here. More details about the handling in the context of SCL can be found later in this standard.

3.1

extensible

a language is extensible if instances of the language can include terms from other vocabularies

Note 1 to entry: This is fulfilled in SCL if the other vocabularies come with their own XML name space.

3.2

language

an identifiable set of vocabulary terms that has defined constraints. For the purpose of this standard, the language is SCL.

Note 1 to entry: This is the case with SCL, although some constraints are not definable in the XML schema.

3.3

instance

a realization by usage of a language

Note 1 to entry: For example, an XML document in SCL describing an IED or a substation is an SCL instance.

3.4

sender

a tool that creates or produces an instance for processing by another application (receiver)

Note 1 to entry: SCL senders are typically IED and system configuration tools; e.g. the IED tool sends (produces) ICD files, the system tool sends SCD files.

3.5

receiver

a tool that consumes an instance which it obtained from a sender

Note 1 to entry: SCL receivers are IED tools and system configuration tools; e.g. the IED tool receives SCD files, the system tool ICD, IID, SSD and SED files.

3.6

processor

a component which receives SCL instances and produces new instances, i.e. is sender and receiver

Note 1 to entry: This is typically the system configuration tool.

3.7

project

a system part with engineering responsibility for all contained IEDs

Note 1 to entry: 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.

3.8

backwards compatible

a language change is backwards compatible, if newer receivers can process all instances of the old language

Note 1 to entry: 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.

3.9

forward compatible

a language change is forward compatible if older receivers can process all instances of the newer language

Note 1 to entry: 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.

3.10

language version

the version of the XML schema defining the language

Note 1 to entry: 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 exclusive to this standard, or particularly useful for understanding this standard 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)
ICT	IED Configuration Tool
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	System Configuration description Language
SCT	System Configuration Tool
SDI	Instantiated Sub-DATA; middle name part of a structured DATA name
SDO	Sub-DATA within a DOType, referencing another DOType
SED	System Exchange Description
UML	Unified Modelling Language according to http://www.omg.org/uml
URI	Universal Resource Identifier
UsvID	ID for USV (Unicast Sampled Value)
XML	Extensible Markup Language

5 Intended engineering process with SCL

5.1 General

Engineering of a substation automation system may start either with the allocation of functionally pre-configured devices to switch yard 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 focused on these purposes:

- 1) SAS functional specification (point 5.1 a) above),
- 2) IED capability description (points 5.1 b) and 5.1 c) above), and
- 3) SA system description (points 5.1 d) and 5.1 e) above).

These purposes shall provide standardized support to system design, communication engineering and to the description of readily engineered system communication for device engineering tools.

For practical purposes, the following is also supported:

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

This is achieved by defining an object model describing the IEDs, their communication connections, and their allocation to the process functions and equipment, as well as 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 carried out in a standardized and compatible way. IEC 61850-4 gives an overview about the complete system life cycle.

5.3 Use of SCL in the engineering process

Figure 1 explains the usage of SCL data exchange in the above-mentioned engineering process. This is just a part of the overall engineering process and the complete system life cycle as described in IEC 61850-4. The text boxes in Figure 1 above the dashed line indicate where SCL files are used. The text box *IED capabilities* corresponds to a result of steps 5.1 b) and 5.1 c) above, the text box *System specification* corresponds to step 5.1 a) above, the text box *Associations...* refers to steps 5.1 d) or 5.1 e) above.

To make the engineering tasks and responsibilities clear, tool roles are introduced for an IED configurator and a system configurator. Clause 10 provides more details about the engineering rights of the tool roles. A

tool implementation can have both roles. In this 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, i.e if the modification was done in the scope of an IED tool, or in the scope of a system tool.

The **IED Configurator (ICT)** 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 claiming conformance to the IEC 61850 series shall fulfil the following:

- 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 (SCT)** 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 of one or several manufacturers, 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 project-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 process part.

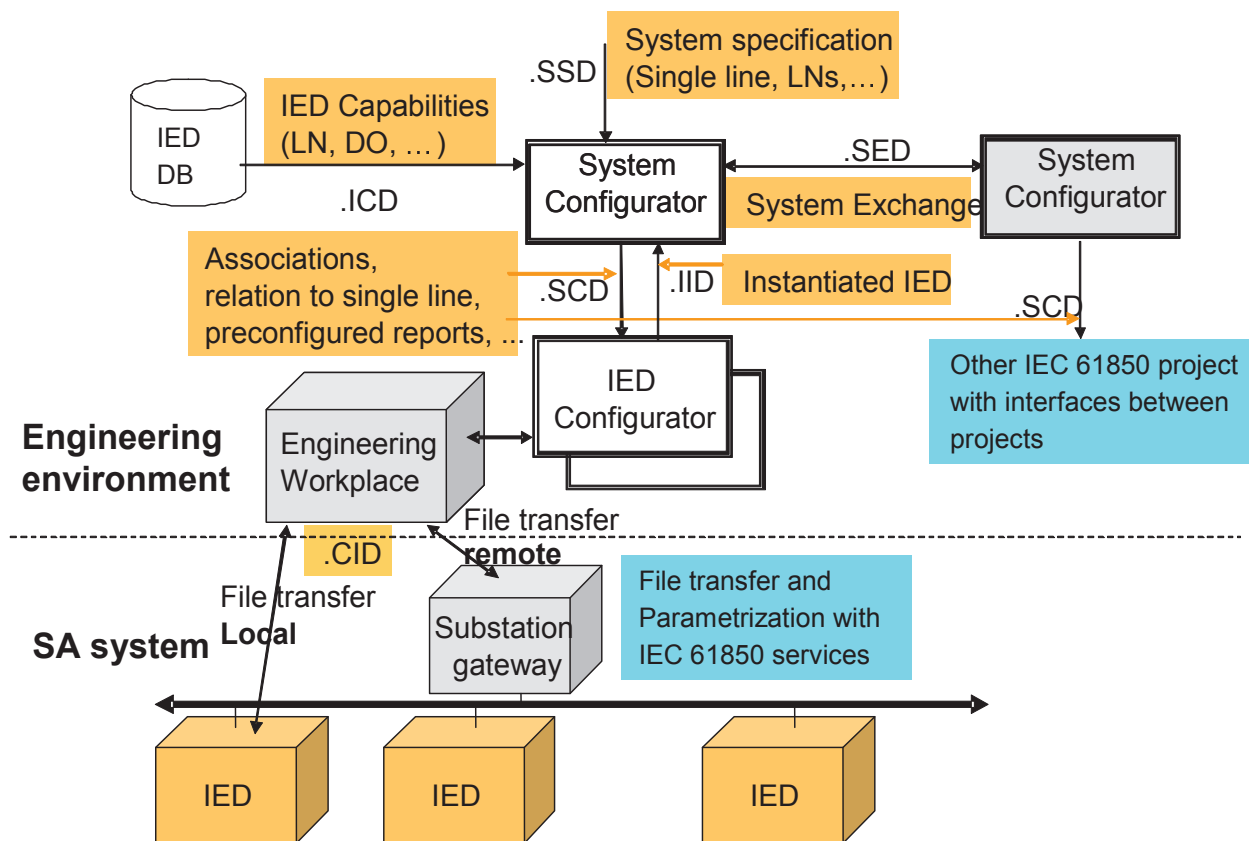


Figure 1 – Reference model for information flow in the configuration process

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 effected 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 standardized 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 as described above. Both the system configurator as well as the IED configurator introduced above are conceptual tools, respectively 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 to support engineers by a specific software tool. In addition, complete freedom of choice is given in 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 not covered by 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.

Figure 1 gives a static view of file type exchange between tool roles. The sequence of data exchanges between engineering tools during the engineering process can then be as follows (see Figure 2):

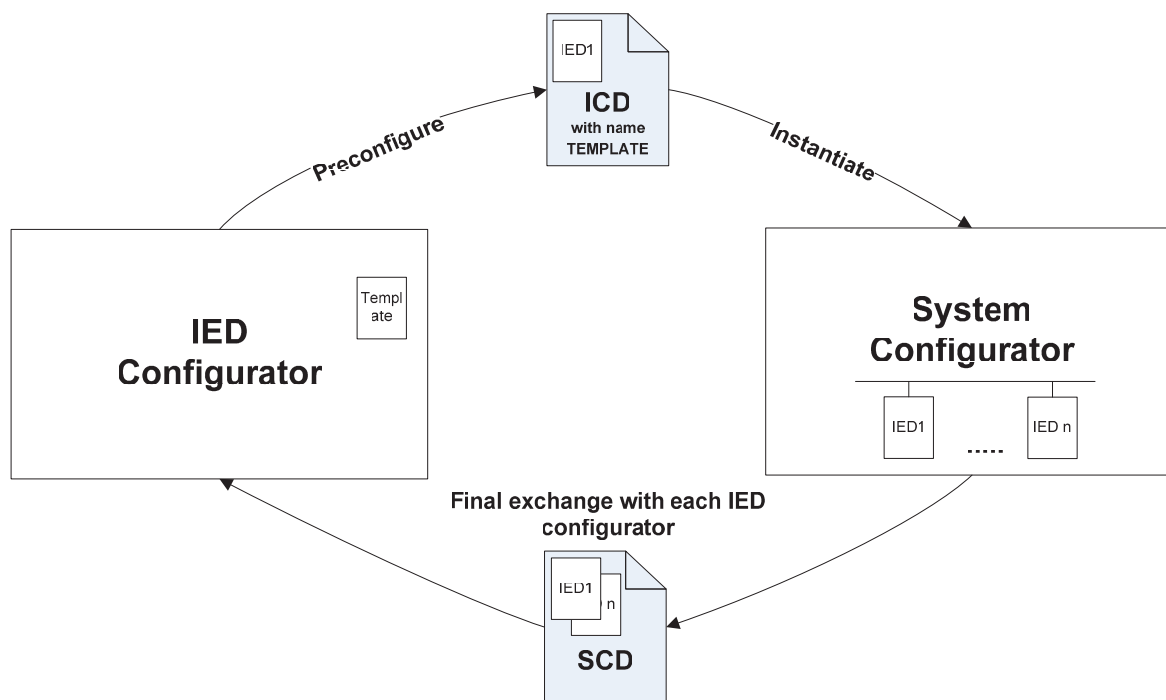


Figure 2 – IED type description to System Configurator

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 subclause on IED modifications.

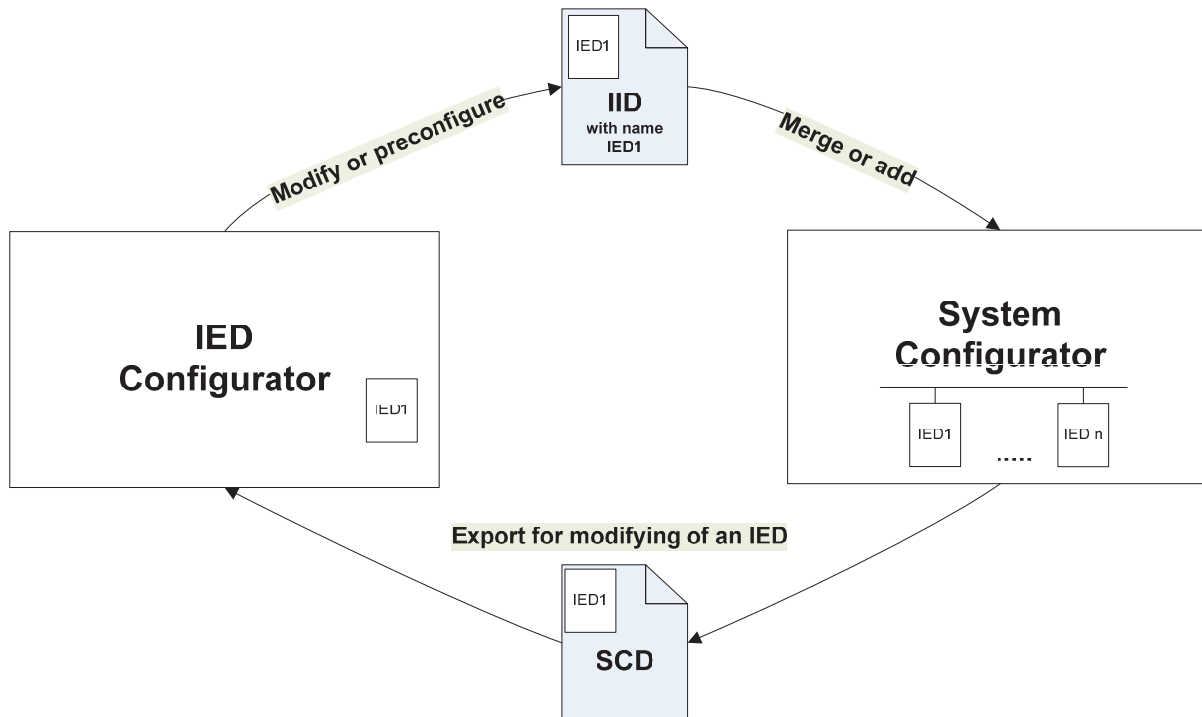


Figure 3 – IED instance description to System Configurator

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 reinstantiating 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 clarified 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 coming from system engineering. To assure this, it shall not 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 the case that this IED does not exist in the SCD file, it can be imported completely and instantiated as a project specific IED, without any references to other IEDs (see also Figure 3). In the 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, shall still be valid. The IED configurator is not allowed to make changes which modify the already configured data flow and communication addresses. 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.

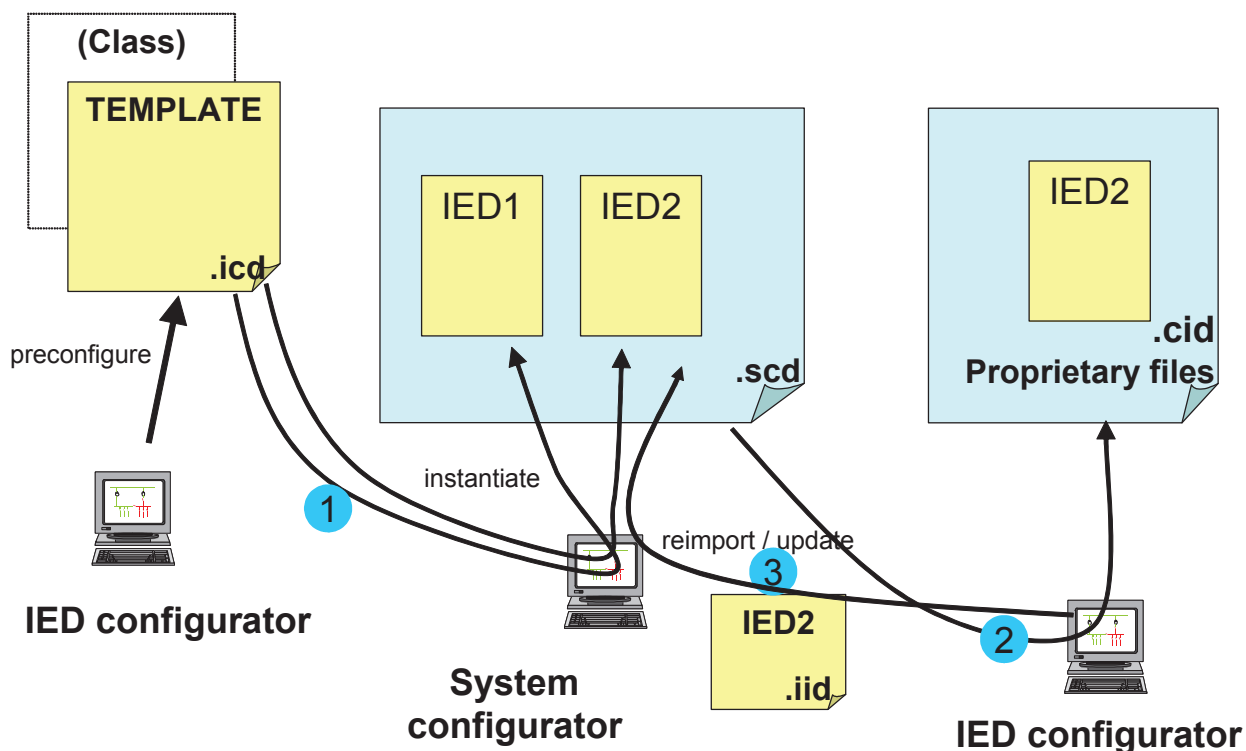


Figure 4 – Modification process

At the start of system engineering 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, the IED owner within the IID file should be set to the header identification of the SCD file, which was the input to IED engineering before.

5.5 Data exchange between projects

As far as the engineering responsibility is concerned, a complete secondary system can be split into different parts. Examples include but are not limited to separate engineering of high-voltage level and medium-voltage level, of a transformer-related part, of separate bays, or even of different substations exchanging data e.g. for line protection or interlocking. For the purposes of this standard, such a system part with responsibility for all its contained IEDs is called a **project**. 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 1 From a static point of view a *project* can be considered as a responsibility area. As the exchange of information between responsibility areas is carried out in processes, the term *project* is used here.

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

- 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.
- A project can transfer to another project the right to add definitions for data flow from some of its IEDs (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.

- c) To not lose already engineered references on such exported IED parts, 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.
- d) Needed parts of the Substation section and communication section shall be exported. From the Substation section only full bays shall be exported, although they shall contain 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. Furthermore, 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 ensure that objects entered by him also have unique names in the exporting project, and that the entered addresses are unique within the full Subnetwork.

The transfer of rights 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 rights shall be frozen for engineering at the IED owner project (after IED check-out), until the using project transfers it back by means of another SED file (IED check-in). The file transferred back shall have the same SCL Header identification as the originally exported file but with an increased revision index.

The Header identification of an 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 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.

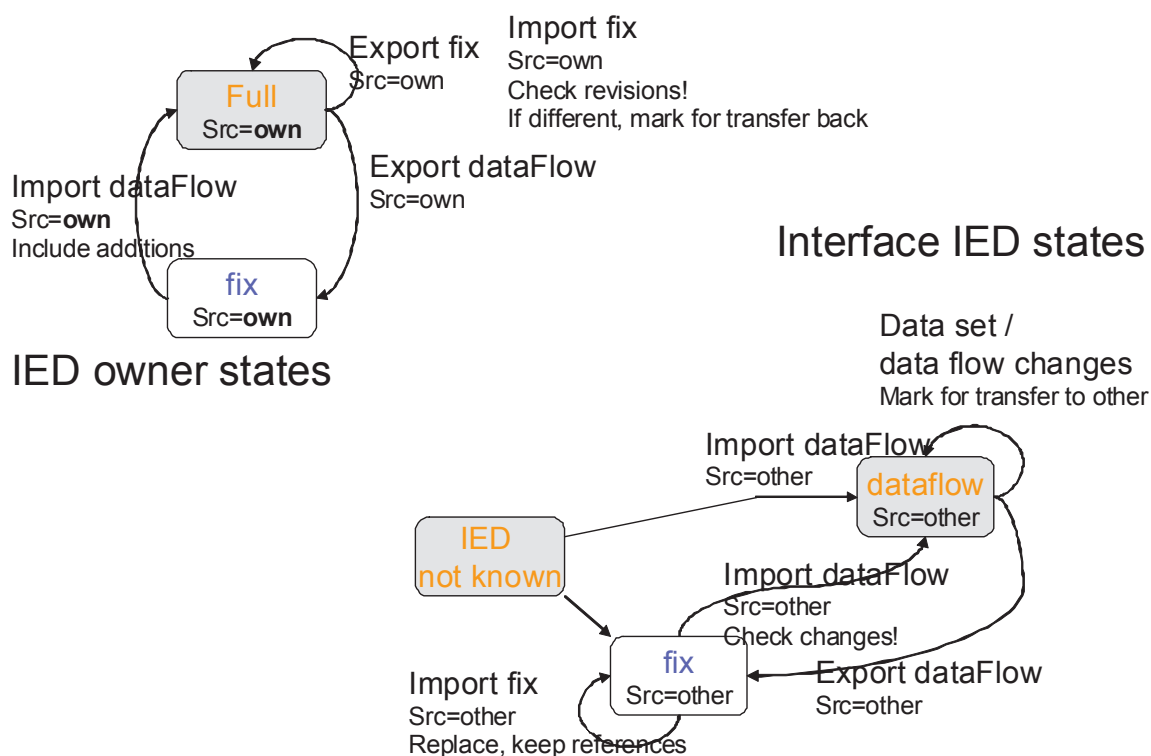


Figure 5 – Engineering right handling in projects

The state diagram on the left 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 the owner of the IEDs. The src property defines whether the IED owner is the own project or some other project.

Example: the owner project exports an IED with engineering right fix. In this case itself contains full engineering rights – however also to inform the other project in case something relevant for it is changed. The other project then imports this IED. As the IED right is fix, it is not allowed to change anything at the IED, however it will use it as destination IED for some of its own IEDs which need to send data to it. When ready, the imported IED as well as any IEDs from the other project sending data to it are exported as SED file and imported in the owner project, which now can complete the IED engineering to receive the data from the other project.

The detailed engineering rights in terms of SCL elements are 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 rights is also shown in the SCD file, if a 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 engineer(s) to use them in such a way, that the system description generatable as an 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 2 The engineering rights 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 language in its full scope allows to describe a model comprising:

- the primary (power or process) system respective function structure: which primary apparatus functions are used, and how the apparatus are connected. This results in a designation of all covered primary equipment like switchgear as utility (automation) functions, structured according to IEC 81346-1;
- the communication system: how IEDs are connected to subnetworks and networks, and at which of their communication access points (communication ports) with which addresses;
- 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 node instances 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; further IED engineering capabilities;
- instantiable logical node (LN) type definitions. The logical nodes as defined in IEC 61850-7-x and other domain standards 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 standard 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 process 7 switch yard (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 61850-7-1. 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.

An 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.

Clause 6 therefore gives an overview of the model by using UML notation. The following clauses then define how an instance of the model is formally described in SCL.

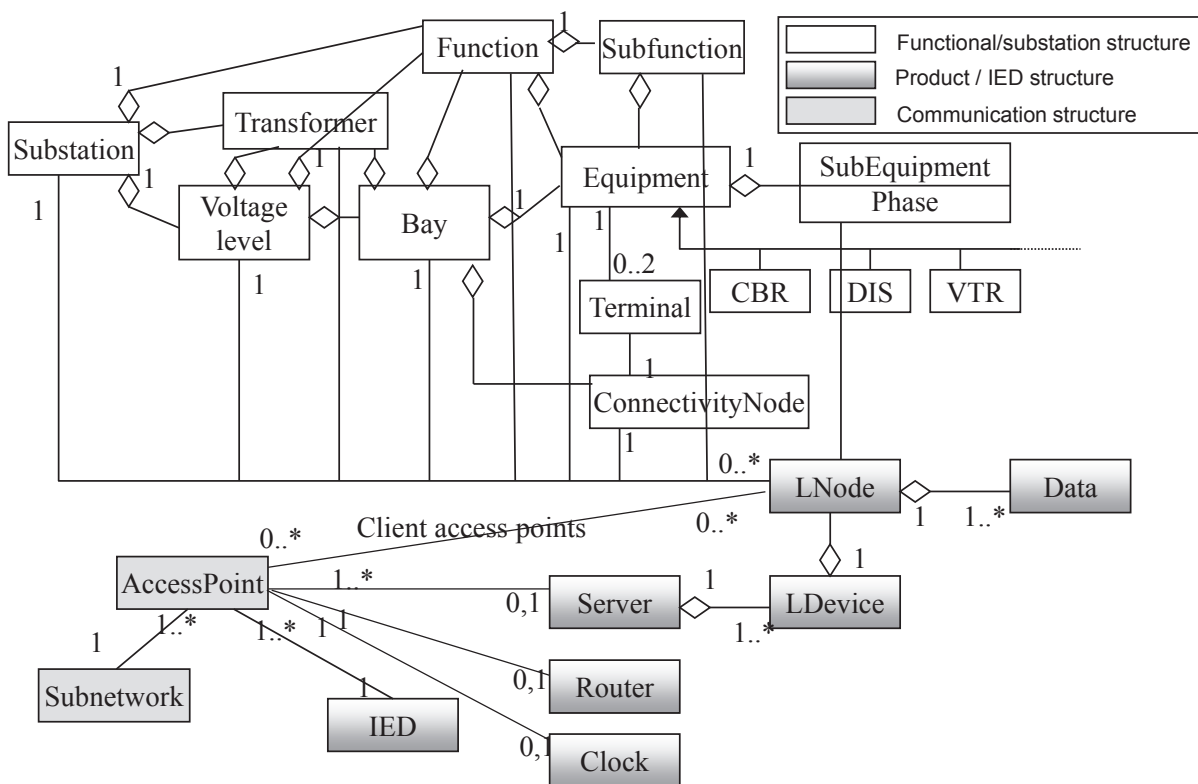
The UML object model is contained in Figure 6. Note that it is not complete in the modelling sense, i.e. it does not show any superclasses from which the used classes may be derived, or any attributes. It restricts itself to those concrete object types that are used within a SCL instance file, in the 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 DataTemplates clause.

The object model has three basic parts:

- 1) Substation / Line / Process: this part describes the primary process related functions and devices like switch yard, respectively any primary process in the functional view according to IEC 81346-1, electrical connections on single line level (topology), and the designation of equipment and functions;
- 2) Product: this stands for all SA product-related objects such as IEDs and logical node implementations;
- 3) Communication: this contains communication-related object types such as subnetworks and communication access points, and describes the communication connections between IEDs as a base for communication paths between logical nodes as clients and servers.

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

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



IEC

Figure 6 – SCL substation object model

The process / 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 81346 (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. 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 makes it possible to specify whether 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 Figure 6 shows, 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 switch yard functionality and a communication aspect as a client or as a server within the substation automation system.

The process / 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 81346-1. The function structure of IEC 81346-1 shall be used, and the designation coding of IEC 81346-2 should be used in the process / substation objects, while the IEC 81346-1 product structure should be used for IED designation structure and the IEC 81346-2 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 subclauses 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 switch yard functionality is however only found in this part of IEC 61850, and therefore shown as far as it is 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 switch yard. The naming is performed according to the IEC 81346 series. The switch yard 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 switch yard functionality (for example the bay controller -E1Q1SB1 as a product is assigned to bay =E1Q1, and its LN CSW11 controls the circuit breaker =E1Q1QA1 via the LN XCBR1 on the IED -E1Q1QA1B1). Observe that in IEC 81346-1 terms, in this instance the bay is a transition object, i.e. it has a function (= sign, at switch yard level), and it is considered to be a part of the switch yard as a product.

This transition can 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.

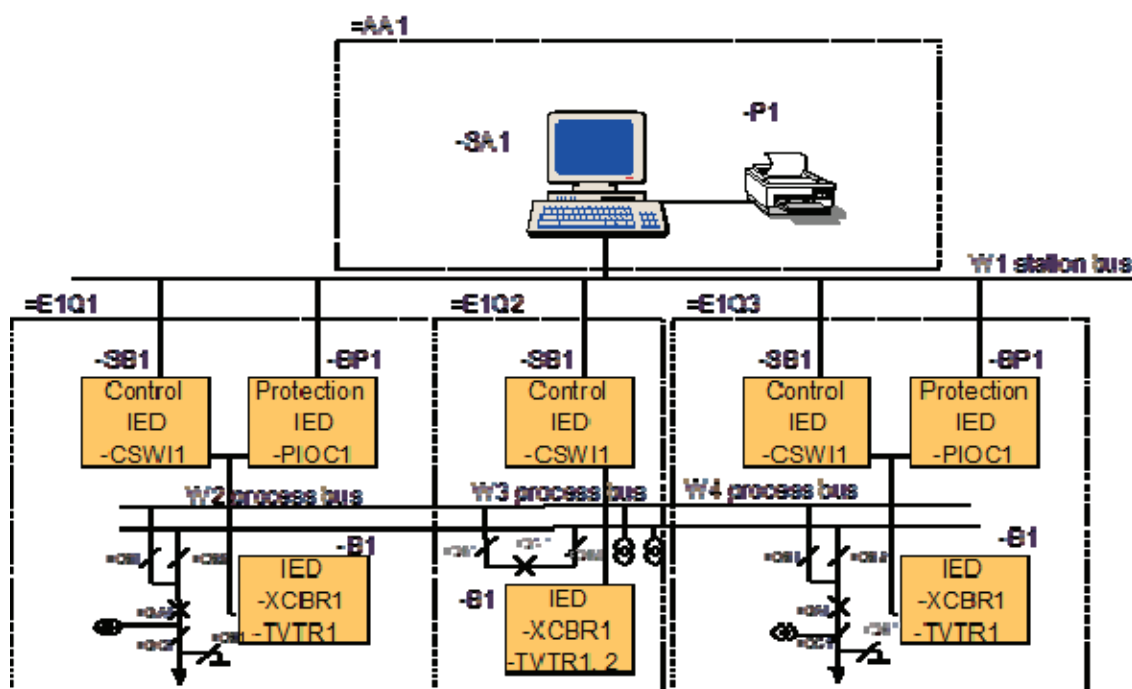


Figure 7 – SA System Configuration example

6.2 The process model

The process model (upper part of Figure 6 for substations) is an object hierarchy based on the functional structure of the primary process. A special primary process is the Substation, which has its own SCL element. Another is the Line, which connects electrically bays of substations. The generic *Process* element shall be used to model any additionally needed structuring levels of a power grid above the substation and for all other primary processes. 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 Process / Substation respective Line hierarchy, they can be attached as functional objects at each 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 process model is

- to relate a logical node and its function to a function of or at the primary process (process part or substation part or line part or equipment or subequipment);
- to derive a functional designation for the logical node from the process structure.

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

Process	the whole or part of the primary process handled by the automation system, which is no substation. Can be parts of the power grid containing several substations, or complete other primary processes like power plants.
Line	a line connecting several substations.
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 switch yard (substation) within one voltage level.
Equipment	an apparatus within the switch yard, for example circuit breaker, disconnecter, voltage transformer, power transformer winding etc. The single line diagram of a switch yard 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 a Function or SubFunction, e.g. earth fault protection as subpart of the main 1 function.
EqFunction	allows additional functions at or below Equipment level, e.g. redundant functions on the same equipment.
EqSubFunction	a hierarchical subpart of an EqFunction or EqSubFunction.

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 needs at least 2 characters, from which within the SCL substation structure the first character is taken as the voltage level name, and the last character is taken as the name for the one bay element.

6.3 The product (IED) model

Products consisting of hardware or software implement the functions of the switch yard. 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 utility automation device performing automation functions by means of logical nodes (LNs). It normally communicates via a communication system with other IEDs in the automation 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) instance 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 configurator 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 only reach 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 SCL, although some features allow to model it at least partly – see also 9.4.6.

This 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.

Furthermore, 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. This type SWITCH is stated by means of the IED type attribute. 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 the bridge level, but no routing on the 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.. An access point may 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 receive data as a client from a process

bus, 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 layer 2 services such as basic GOOSE and sampled value 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 IEC 61850-7-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: If the communication system is doubled, but below the addressing level provided for a logical access point, this can be described in SCL at the level of physical connections of an access point. There might be additional SCSSM specific parameters or application level supervision data, if the redundancy issue is taken up in the stack mapping. Other communication system redundancy mechanisms can only be described at physical level. 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. However SCL provides some optional means to describe the physical connections at port / cable level also for rings.
- 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 modelling

Conceptually the IEC 61850 data flow has logical nodes on servers or publishers as source, and logical nodes as clients respectively subscribers. The real connections/associations however are built at communication profile level (e.g. MMS/TCP), and these 'association channels' can be assigned to a client/subscriber IED, an access point, a logical device or – as in the IEC 61850 base model – a logical node. If the channel is built by an IED, then all LNs hosted in this IED can use it in their client role. 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 the data flow to be modelled 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, such as in the original IEC 61850 client model. If in this case several client LNs share the same channel, it is recommended taking the LLN0 as the client LN, because LLN0 represents a whole logical device.

At data object 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 an SCL level, or, if the IED supports this, even on the IEDs LN data model by means of data objects of CDC ORG (see IEC 61850-7-4). Also here it is often the case that the same incoming data object 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.

One of the big advantages of IEC 61850 is that the communication-related data flow is defined on top, but independent from the application level data model. To make it easier for an engineer to understand and define this data flow, the SCL language restricts the definitions in 7-2 as follows: data set definitions referenced by a control block must be in the same logical node as the control block. This means automatically, that all GOOSE and SMV data flow definitions are in LLN0. It is recommended, if the IEDs allow this, to also keep report data flow definitions there. Any online changes not following this convention cannot be documented in SCL language.

7 SCL description file types

SCL files are used to exchange the configuration data between different tools (conceptual tool roles as defined in clause 5), possibly implemented by different manufacturers. As already mentioned in subclause 5.1 (see also Figure 1), there are at least six different purposes for SCL data exchange, and therefore six 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 System Configuration description Language (SCL) defined in the next clause. 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, IEC 61850-7-4 or is a private issue of the tools.

The following types of SCL files are distinguished:

- Data exchange from the IED configurator to the system configurator (corresponding to items b) and c) of 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 process, line or substation section, where the highest level name shall be **TEMPLATE**. If a process TEMPLATE is defined, the binding of logical node instances to primary equipment indicates a predefined functionality. Any process in which this IED shall be used, must match an appropriate process topology part (example: a CSWI LN bound to an equipment of type CBR is only allowed to control a circuit breaker; a CILO bound to a line disconnector implements the interlocking logic for a line disconnector). There might be an optional Communication section defining possible default addresses of the IED. A specific SCSM might make this mandatory for some address parts.

The file extension shall be ICD for IED Capability Description.

- Data exchange from the IED configurator to the system configurator 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, it may also have project specific addresses, and a data model possibly included with 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 a data set and control block definitions, which must either be identical to those in the system tool in case modifications are transferred after system engineering, 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 for Instantiated IED Description.

- Data exchange from a system specification tool to the system configurator. 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 an LN in the substation section is not bound to an IED and also has no logical node type defined, then only the mandatory part of this LN according to IEC 61850-7-4 is specified. If part of the SA system is already known, this might optionally be contained in IED and Communication sections.

The file extension shall be SSD for System Specification Description.

- Data exchange from the system configurator to IED configurators (corresponding to items d) and e) of subclause 5.1). This file contains all IEDs including the configured data flow and needed *DataTypeTemplates*, a communication configuration section and a substation description section.

The file extension shall be SCD for System Configuration Description.

- Data exchange from the IED configurator only 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 a vendor specific SCD file, possibly stripped down to what the concerned IED shall know (restricted vendor specific view of IEDs). If a compression method is applied, those according to RFC 1952 is 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 the form of IEC 61131-3 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 configurators 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 lose the source object of 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 / publisher or client / subscriber according to the IEC 61850 standard shall be accompanied by an ICD file, respectively by a tool capable of generating an ICD file, or a project specific IID file, respectively a tool capable of generating a project specific IID file for this IED, and shall be able to consume an SCD file or 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 or 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 IEC 61850-7-4, which allows a lot of functionality to run on the IED hardware, however not all at once. Each ICD file then is a runnable (implementable) subset of all possibilities of the IED class. Only where all available functions and function instances can run on the IED hardware, will there exist only one ICD file. This issue is illustrated in Figure 8 for the most general case.

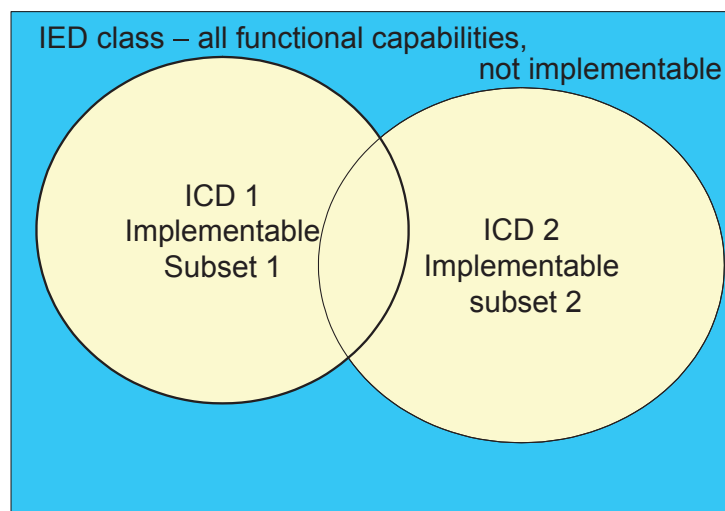


Figure 8 – ICD files describing implementable IED types of a general IED class

8 SCL language

8.1 Specification method

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

The 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 a XML schema. Constraints on the object model which are not or not easily able to be formulated 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 adding *Private* sections and a descriptive *Text* to the element. It also allows adding 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. An example can be found in Annex C.1.

The next level of element types is based on *tBaseElement*:

- *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. Attribute *name* is of type *tName*, i.e. also strings that do not contain any carriage return, line feed, or tab character, but cannot be empty. Attribute *id* of type *tID* is more restrictive than *tName*: any whitespace character is not accepted and size is limited to 255 characters.

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 re-use, the whole SCL schema is split into 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 process (Process, Line, 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:

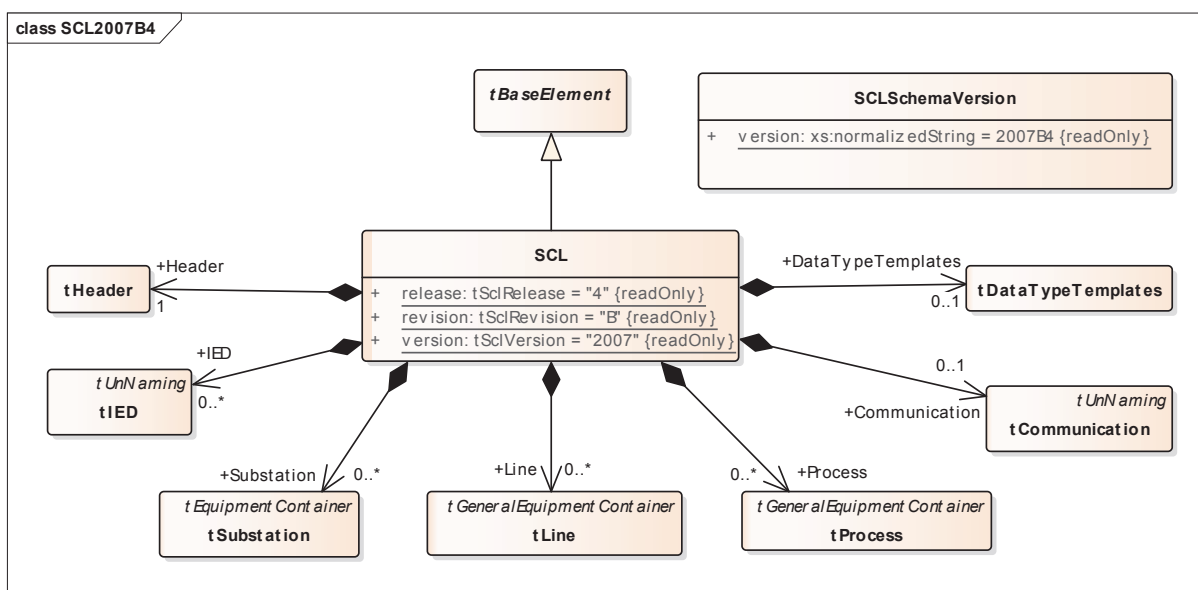
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.iec.ch/61850/2003/SCL"
  xmlns:scl="http://www.iec.ch/61850/2003/SCL"
  xmlns="http://www.iec.ch/61850/2003/SCL"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  finalDefault="extension" version="9999A9">
```

where version 9999A9 is the year, index letter and release number stating the SCL schema version, which is 2007B4 for this document. 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.



IEC

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 a process description with one of the elements process of type *tProcess*, Line of type *tLine*, and Substation elements of type *tSubstation*, a Communication section of type *tCommunication*, one or more IED elements of type *tIED*, and a DataTypeTemplates section of type *tDataTypeTemplates*. All these element types are handled in later clauses; the usage of Line and Process elements might be detailed in other parts of the standard.

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. SCL may use values to identify another element of the XML file. When doing this, the referenced element shall exist in the file, otherwise the structure is invalid. For example (full definition of this example is given later) in element GSE, a specific attribute called *cbName* indicates the element GSEControl and attribute *IdInst* indicates the Ldevice containing this GSEControl, for which the GSE has been created, and if this GSEControl is not present, the GSE shall not exist. 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 61850-9-2. 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 they need a value, they shall have default values, whose meaning is as far as possible 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 backward compatibility as well as forward compatibility, the *may ignore* and *must understand* rules are introduced into the language definition. These allow having different language versions for the same SCL language name space.

8.2.1 MustUnderstand rules

The MustUnderstand / MayIgnore rules have to be followed when reading an SCL file. When producing SCL output, this should conform to the claimed supported SCL version/revision/release.

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. If a known element contains directly below it an element with *mustUnderstand* property which a tool does not understand, then it must also ignore the known containing element.

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 configurator and the IED configurator 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 *DataTemplate* 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. The following example illustrates the use of the *mustUnderstand* feature.

An IED defines a GOOSE control block as follows:

```
<GSEControl name="GoCB02" appID="GoCB02" datSet="CmdResv">
  <IEDName>AA1_D1_Q11A1</IEDName>
  <IEDName>AA1_D1_Q10A1</IEDName>
  <IEDName>AA1_D1_Q07A1</IEDName>
  <Protocol mustUnderstand="true">R-GOOSE</Protocol>
</GSEControl>
```

If the tool importing this does not understand the *Protocol* element, then it is also not allowed to use the *GSEControl* element containing it (e.g. add IED names), although *GSEControl* is known to it.

Tools should 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 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 B. If error-correcting corrigenda of this standard follow before any new version of this standard is published, the first corrigendum will get the identification B, the next C, etc. Fixing of interoperability related errors in between revisions is indicated by a release number. The first released revision will always be release 1. The release belonging to this standard is 3.

The special XML schema for this current edition of IEC 61850-6 is contained in Annex A, 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.

For backward compatibility all tools have to accept SCL instances from older as well as newer versions, including the features deprecated in the newer version(s) as far back as declared with the tool version. Therefore a tool input cannot be verified against the schema of the version defined in this standard, but at best against the schema version with which the input instance is produced. The general schema in Annex E gives a hint as to what shall be tolerated by a tool supporting the valid 2003 version as well as this current 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 a '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, or to implement the upgrading / downgrading rules from Annex I.

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 schema version as defined in Subclause 8.1. However, not all schema versions will be released, and SCL language versions, even bug fix versions, will only be defined for released / published SCL schema versions.

8.2.3 Incompatibilities to earlier versions

This current version of SCL with version identification 2007 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; it is currently used for the *Protocol* element contained in GOOSE and SV control block definitions.
- The order where the Log element appears has changed: it shall appear directly before any control block definitions which belong only to LN0 (like GOOSE control blocks).
- If the log control block does not reside in the same logical device as the log, the *IdInst* attribute shall be stated explicitly.
- The access point name allows only alphanumeric characters and underscore (_). Its length is restricted to 32 characters.
- The newly introduced LDevice attribute *IdName* leads to incompatibilities with edition 1 (2003A) tools, as ignoring this attribute leads to wrong communication level configuration.
- The attribute *ReportControl.rptId* shall no longer have the empty string value.
- It is clarified that the meaning of the *maxAttributes* attribute of the *ConfDataSet* element denotes FCDAs and not basic attributes.
- The modeling of the transformer neutral point has been corrected
- The *max* attribute of the *SupSubscription* element has been replaced by two special attributes for GOOSE and SV supervision.
- The length of *DataTypeTemplate* identifiers is restricted to 255 characters.
- *EqFunction* and *EqSubFunction* are used below Equipment elements instead of *Function* / *SubFunction*
- The length of Enumeration strings is restricted to 127 characters of ISO Latin and Latin-1 supplement characters
- The SubNetwork type code is mandatory and not the empty string, except in SSD files.
- The value of *RptEnabled.max*, if defined, shall be >0.
- The meaning of missing *originalSclVersion* / *originalSclRevision* attributes is SCL version/revision 2003A
- The string 'None' is forbidden as an IED name. It is only allowed as an IED name substitute inside the Process / Substation section to specify logical nodes without relation to an IED.

It is recommended to fix these issues together with the implementation of the *mustUnderstand* property also in 'old' tools. Then they can handle all future SCL versions. For compatibility between versions see also Annex I.

8.3 SCL language extensions

8.3.1 General

The SCL language elements without those serving extension purposes are designed for a specific purpose as 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, 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 objects 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 inheriting from *tBaseElement*, (sub-)tags and attributes can be added. These shall however belong to a defined XML name space with defined semantics for all these elements. It is recommended to define the used name spaces at the main tag (SCL), although standard XML allows to add it also below at elements from this name space. Observe that standard XML rules allow to replace this definition.

This namespace shall not be the same as the target namespace of the SCL schema (see below). For private name spaces, the used internal name space abbreviation should start with the character *e*. IED configurator tools shall be aware that the used name space abbreviation might be changed by the system configurator, if its relation to the referenced URI is not unique across all used ICD/IID files. 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 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 SCL elements of 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 cannot be the SCL name space. The Private element allows also to reference other files by means of a URL at its *source* attribute.

The handling within tools shall be as follows:

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

Private data for different purposes shall be distinguished by the value of its *type* attribute. If manufacturers use a Private definition, 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 in order 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).

Private 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.

NOTE Due to the engineering process as described in clause 5 IED configurators should not put Private elements into the Communication section or the DataTypeTemplate section.

8.3.7 Another XML syntax

A completely new standardized 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 language elements as a default XML name space; understand all parts of the syntax referring to the capabilities of the handled IEDs and the intended functionality of the tool; ignore all SCL language elements which it does not understand, following the *mayIgnore* / *mustUnderstand* principle (see 8.2.1);
- 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 are 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) other 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 and shall be supported if standard conformance is claimed. The XML encoding attribute shall be processed in a case-insensitive way. SCL generators shall always use upper case encoding.

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="B" release="3">
```

```
<!-- 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.

For an XML processor, this example 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 needs to be carried out.

The SCL element shall contain a header section, and at least one of the following sections: Process or Substation, Communication, IED, DataTypeTemplates, which are further explained below. The Process/Substation and the IED sections may appear more than once. Figure 9 gives an overview as an UML diagram.

Most 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.

All SCL level references to objects on an IED use the IED-related names, i.e. the IED name and LD instance name, even if at communication level other identifications might be used. This is valid for references from the substation section to logical nodes on the IED, but also for references within an IED, e.g. to define data objects which are the members of data sets.

Observe further that the SCL element has the attributes *version* with value 2007 for this version of the SCL language, *revision* with value B for this revision of the 2007 language version, and *release* with value 3 for this published release.

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