



IEC 61850-7-1

Edition 2.1 2020-08
CONSOLIDATED VERSION

INTERNATIONAL STANDARD



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





THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2020 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and definitions clause of IEC publications issued between 2002 and 2015. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.



IEC 61850-7-1

Edition 2.1 2020-08
CONSOLIDATED VERSION

INTERNATIONAL STANDARD



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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 33.200

ISBN 978-2-8322-8823-8

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD	11
INTRODUCTION	13
1 Scope	14
2 Normative references	14
3 Terms and definitions	15
4 Abbreviated terms	16
5 Overview of the IEC 61850 series concepts	17
5.1 Objective	17
5.2 Topology and communication functions of substation automation systems	18
5.3 The information models of substation automation systems	19
5.4 Applications modelled by logical nodes defined in IEC 61850-7-4	21
5.5 The semantic is attached to data	24
5.6 The services to exchange information	26
5.7 Services mapped to concrete communication protocols	28
5.8 The configuration of the automation system	28
5.9 Summary	29
6 Modelling approach of the IEC 61850 series	30
6.1 Decomposition of application functions and information	30
6.2 Creating information models by stepwise composition	32
6.3 Example of an IED composition	34
6.4 Information exchange models	35
6.4.1 General	35
6.4.2 Output model	36
6.4.3 Input model	40
6.4.4 Model for statistical and historical statistical data	50
6.4.5 Model for system functions	53
7 Application view	55
7.1 General	55
7.2 First modelling step – Logical nodes and data	57
7.3 Mode and behaviour of a logical node	60
7.4 Use of measurement ranges and alarms for supervision functions	60
7.5 Data used for limiting the access to control actions	61
7.6 Data used for blocking functions described by logical nodes	61
7.7 Data used for logical node inputs/outputs blocking (operational blocking)	61
7.7.1 General	61
7.7.2 Blocking incoming commands	62
7.7.3 Blocking process outputs	62
7.7.4 Blocking the communication of status outputs updates	62
7.8 Data used for testing	63
7.8.1 General	63
7.8.2 Multicast signals used for simulation	63
7.8.3 Input signals used for testing	64
7.8.4 Test mode	66
7.9 Logical node used for extended logging functions	66
8 Device view	67

8.1	General.....	67
8.2	Second modelling step – logical device model	68
8.2.1	The logical device concept.....	68
8.2.2	The device nameplate.....	69
8.2.3	Gateways and proxies.....	70
8.2.4	Logical devices for monitoring external device health.....	74
8.2.5	Logical devices management hierarchy.....	74
9	Communication view.....	76
9.1	General.....	76
9.2	The service models of the IEC 61850 series	76
9.3	The virtualisation	78
9.4	Basic information exchange mechanisms	79
9.5	The client-server building blocks.....	81
9.5.1	Server.....	81
9.5.2	Client-server roles	82
9.6	Logical nodes communicate with logical nodes	83
9.7	Interfaces inside and between devices.....	84
10	Where physical devices, application models and communication meet.....	85
11	Relationships between IEC 61850-7-2, IEC 61850-7-3 and IEC 61850-7-4	86
11.1	Refinements of class definitions.....	86
11.2	Example 1 – Logical node and data object class	87
11.3	Example 2 – Relationship of IEC 61850-7-2, IEC 61850-7-3, and IEC 61850-7-4 ..	91
12	Formal specification method	93
12.1	Notation of ACSI classes	93
12.2	Class modelling	93
12.2.1	Overview	93
12.2.2	Common data class	95
12.2.3	Logical node class	99
12.3	Service parameters tables	101
12.4	Referencing instances	102
13	Namespaces.....	105
13.1	General.....	105
13.2	Namespaces defined in the IEC 61850-7-x series	106
13.3	Namespaces dependencies	110
13.3.1	General.....	110
13.3.2	Basic namespaces dependencies	110
13.3.3	Other namespaces dependencies	112
13.4	Specification of namespaces.....	113
13.5	Attributes for references to namespaces	113
13.5.1	General.....	113
13.5.2	Attribute for logical device namespace (ldNs)	114
13.5.3	Attribute for logical device basic namespace (LLN0.NamPit.InNs).....	114
13.5.4	Attribute for logical node namespace (lnNs).....	115
13.5.5	Attribute for data namespace (dataNs).....	115
13.5.6	Attribute for common data class namespace (cdcNs)	115
13.5.7	Attribute for naming a common data class name (cdcName)	115
13.6	Deprecation of namespaces.....	116
14	Common rules for new version of classes and for extension of object classes	116

14.1	General.....	116
14.2	Basic rules.....	116
14.2.1	General.....	116
14.2.2	Use of optional information.....	119
14.2.3	General rules for enumerations types and enumeration-based data objects and attributes.....	119
14.3	Rules for extensions within private namespace.....	119
14.3.1	Rules for LN classes.....	119
14.3.2	New LN classes.....	120
14.3.3	Rules for Common Data Classes.....	120
14.3.4	Rules for enumeration types and enumeration-based data objects and attributes.....	121
14.4	Extensions made within product standard namespaces.....	121
14.5	Extensions made within transitional namespaces.....	121
14.6	Extensions made within basic and domain namespaces.....	121
14.7	Multiple instances of LN classes for dedicated and complex functions.....	122
14.7.1	Example for time overcurrent.....	122
14.7.2	Example for PDIS.....	122
14.7.3	Example for power transformer.....	122
14.7.4	Example for auxiliary network.....	122
14.8	Specialisation of data by use of number extensions.....	123
14.9	Examples for new LNs.....	123
14.10	Example for new Data.....	123
15	Compatibility between different versions of the standard.....	124
Annex A	(informative) Overview of logical nodes and data.....	125
A.1	Compatible logical node classes and data classes (IEC 61850-7-4).....	125
A.1.1	List of LN groups (IEC 61850-7-4).....	125
A.1.2	LN classes (IEC 61850-7-4).....	125
A.1.3	Data object classes (IEC 61850-7-4).....	125
A.2	Common data class specifications (IEC 61850-7-3).....	125
Annex B	(informative) Allocation of data to logical nodes.....	127
Annex C	(informative) Use of the substation configuration language (SCL).....	130
C.1	General.....	130
C.2	SCL and options in logical nodes.....	130
C.3	SCL and options in data.....	131
Annex D	(xxx).....	132
Annex E	(informative) Relation between logical nodes and PICOMs.....	133
Annex F	(informative) Mapping the ACSI to real communication systems.....	134
F.1	General.....	134
F.1.1	Mapping example (IEC 61850-8-1).....	136
Annex G	(normative) LGOS/LSVS engineering.....	142
G.1	General.....	142
G.2	LGOS/LSVS engineering by the ICT.....	143
G.3	LGOS/LSVS engineering by the SCT.....	144
Annex H	(normative) GOOSE/SMV Subscription Configuration.....	146
H.1	General.....	146
H.2	SCT supplied input binding workflow.....	147
H.3	ICT supplied input binding (for later binding).....	148

Annex I (informative) Specification of namespaces for Edition 3	149
I.1 Namespaces dependencies	149
I.1.1 General.....	149
I.1.2 Namespaces dependencies	149
I.2 Example 2 – Standardized data objects used in standardized LNs	150
I.3 Example 3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B	151
I.4 Example 4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B	152
I.5 Example 5 – Edition 1 device: Logical nodes in technical reports	153
I.6 Example 6- Edition 2.1 device: Logical nodes in technical reports defining new CDCs.....	155
I.7 Example 7- Edition 2.1 device: devices for product family standards.....	156
I.8 Example 8 – Standardized logical nodes introduced by other domain applications	157
I.9 Example 9 – Standardized logical nodes introduced by other domain applications than moved to IEC 61850-7-4	159
Annex J (normative) Use case scenarios examples for clarifying the common rules of Clause 14	162
J.1 General.....	162
J.2 Example 1 – Private LN using standardized DOs	162
J.3 Example 2 – Standardized data objects used in standardized LNs	163
J.4 Example 3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B	163
J.5 Example 4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B	164
J.6 Example 5 – Edition 1 device: Logical nodes in technical reports	165
J.7 Example 6- Edition 2.1 device: Logical nodes in technical reports defining new CDCs.....	167
J.8 Example 7- Edition 2.1 device: devices for product family standards.....	168
J.9 Example 8 – Standardized logical nodes introduced by other domain applications	169
J.10 Example 9 – Standardized logical nodes introduced by other domain applications than moved to IEC 61850-7-4	171
Annex K (normative) General requirements and recommendations regarding compatibility issues between different versions of IEC 61850.....	174
K.1 Overview.....	174
K.1.1 General.....	174
K.1.2 Definitions (to go to part 2):	174
K.1.3 Introduction to compatibility discussion	175
K.1.4 Assumptions	175
K.1.5 Generic rules	176
K.1.6 Overview on use cases	177
K.2 Use cases related to the data model	181
K.2.1 Use case 1: Add new type	181
K.2.2 Use case 2: Add new FC.....	181
K.2.3 Use case 3: Extend CDC with elements of existing types and FCs.....	182
K.2.4 Use case 4: Add a new DataObject (DO) based on new CDCs.....	183
K.2.5 Use case 5: Add a new DO based on existing CDCs.....	184
K.2.6 Use case 6: Rename a DO.....	184
K.2.7 Use case 7: Rename a DA, subDO or subDA	184

K.2.8	Use case 8: Deprecation of a FC	185
K.2.9	Use case 9: Deprecation of a DA	186
K.2.10	Use case 10: Removal of a DA	186
K.2.11	Use case 11: Deprecation of a DO	187
K.2.12	Use case 12: Use a weaker presence condition	187
K.2.13	Use case 13: Use a stronger presence condition	188
K.2.14	Use case 14: Extend Enumeration List with an enumerated value	188
K.2.15	Use case 15: Modification of an Enumerated value	189
K.2.16	Use case 16: Deprecation of an Enumerated value	190
K.2.17	Use case 17: Extension of a PACKED LIST	190
K.2.18	Use case 18: Extend Name Length	190
K.3	Use cases related to the services I.....	191
K.3.1	Use case 30: Add new type.....	191
K.3.2	Use case 31: Using new control block class	192
K.3.3	Use case 32: Using new services.....	192
K.3.4	Use case 33: Extending control block class with an attribute of existing type	193
K.3.5	Use case 34: Rename a service parameter not associated with a control block attribute	193
K.3.6	Use case 35: Removing / Deprecating Control Block Classes	193
K.3.7	Use case 36: Using a stronger presence condition.....	194
	Bibliography.....	195
	Figure 1 – Relations between modelling and mapping parts of the IEC 61850 series	17
	Figure 2 – Sample substation automation topology	19
	Figure 3 – Modelling approach (conceptual).....	20
	Figure 4 – Logical node information categories	23
	Figure 5 – Build-up of devices (principle)	23
	Figure 6 – Position information depicted as a tree (conceptual)	24
	Figure 7 – Service excerpt	27
	Figure 8 – Example of communication mapping	28
	Figure 9 – Summary	30
	Figure 10 – Decomposition and composition process (conceptual).....	31
	Figure 11 – XCBR1 information depicted as a tree.....	33
	Figure 12 – Example of IED composition.....	35
	Figure 13 – Output and input model (principle).....	36
	Figure 14 – Output model (step 1) (conceptual)	37
	Figure 15 – Output model (step 2) (conceptual)	38
	Figure 16 – GSE output model (conceptual).....	38
	Figure 17 – Setting data (conceptual)	39
	Figure 18 – Input model for analogue values (step 1) (conceptual)	41
	Figure 19 – Range and deadbanded value (conceptual).....	42
	Figure 20 – Input model for analogue values (step 2) (conceptual)	43
	Figure 21 – Reporting and logging model (conceptual).....	44
	Figure 22 – Data set members and reporting	45
	Figure 23 – Buffered report control block (conceptual)	46

Figure 24 – Buffer time	47
Figure 25 – Data set members and inclusion-bitstring	48
Figure 26 – Log control block (conceptual).....	48
Figure 27 – Peer-to-peer data value publishing model (conceptual)	49
Figure 28 – Conceptual model of statistical and historical statistical data (1)	51
Figure 29 – Conceptual model of statistical and historical statistical data (2)	53
Figure 30 – Concept of the service tracking model – Example: control service tracking.....	55
Figure 31 – Real world devices	56
Figure 32 – Logical nodes and data (IEC 61850-7-2)	57
Figure 33 – Simple example of modelling.....	58
Figure 34 – Basic building blocks	58
Figure 35 – Logical nodes and PICOM	59
Figure 36 – Logical nodes connected (outside view in IEC 61850-7-x series).....	59
Figure 37 – Mode and behaviour data (IEC 61850-7-4).....	60
Figure 38 – Data used for limiting the access to control actions (IEC 61850-7-4)	61
Figure 39 – Data used for logical node inputs/outputs blocking (IEC 61850-7-4)	62
Figure 40 – Data used for receiving simulation signals.....	63
Figure 80 – GOOSE subscription supervision state machine	64
Figure 81 – SV subscription supervision state machine.....	64
Figure 41 – Example of input signals used for testing	65
Figure 42 – Test mode example.....	66
Figure 43 – Logical node used for extended logging functions (GLOG)	67
Figure 44 – Logical device building block.....	68
Figure 45 – Logical devices and LLN0/LPHD	69
Figure 46 – The common data class DPL.....	70
Figure 47 – Logical devices in proxies or gateways.....	72
Figure 79 – Logical devices in proxies or gateways (functional naming).....	73
Figure 48 – Logical devices for monitoring external device health	74
Figure 49 – Logical devices management hierarchy	75
Figure 50 – ACSI communication methods.....	77
Figure 51 – Virtualisation	79
Figure 52 – Virtualisation and usage	79
Figure 53 – Information flow and modelling.....	80
Figure 54 – Application of the GSE model.....	80
Figure 55 – Server building blocks	81
Figure 56 – Interaction between application process and application layer (client/server)	82
Figure 57 – Example for a service.....	82
Figure 58 – Client/server and logical nodes	82
Figure 59 – Client and server roles	83
Figure 60 – Logical nodes communicate with logical nodes.....	84
Figure 61 – Interfaces inside and between devices	85
Figure 62 – Component hierarchy of different views (excerpt).....	86

Figure 63 – Refinement of the DATA class.....	87
Figure 64 – Instances of a Data object class (conceptual).....	91
Figure 65 – Relation between parts of the IEC 61850 series	92
Figure 66 – Abstract data model example for IEC 61850-7-x.....	94
Figure 67 – Relation of TrgOp and Reporting.....	99
Figure 68 – Sequence diagram	102
Figure 69 – References.....	102
Figure 70 – Use of FCD and FCDA	103
Figure 71 – Object names and object reference	105
Figure 72 – Definition of names and semantics	106
Figure 73 – Name space as class repository	107
Figure 74 – All instances derived from classes in a single name space	108
Figure 75 – Instances derived from multiple namespaces	109
Figure 76 – Inherited namespaces	109
Figure 77 – Basic namespaces dependencies.....	111
Figure 78 – Other namespaces dependencies.....	112
Figure 78 – Basic extension rules diagram.....	117
Figure B.1 – Example for control and protection LNs combined in one physical device	127
Figure B.2 – Merging unit and sampled value exchange (topology)	128
Figure B.3 – Merging unit and sampled value exchange (data)	128
Figure C.1 – Application of SCL for LNs (conceptual)	130
Figure C.2 – Application of SCL for data (conceptual).....	131
Figure E.1 – Exchanged data between subfunctions (logical nodes)	133
Figure E.2 – Relationship between PICOMS and client/server model	133
Figure F.1 – ACSI mapping to an application layer.....	134
Figure F.2 – ACSI mappings (conceptual).....	135
Figure F.3 – ACSI mapping to communication stacks/profiles	136
Figure F.4 – Mapping to MMS (conceptual).....	136
Figure F.5 – Mapping approach	137
Figure F.6 – Mapping detail of mapping to a MMS named variable.....	138
Figure F.7 – Example of MMS named variable (process values)	138
Figure F.8 – Use of MMS named variables and named variable list.....	139
Figure F.9 – MMS information report message.....	140
Figure F.10 – Mapping example.....	141
Figure G.1 – LGOS/LSVS engineering	143
Figure G.2 – LGOS/LSVS engineering by ICT.....	144
Figure G.3 – LGOS/LSVS engineering by the SCT.....	145
Figure H.1 – GOOSE/SMV subscription engineering workflow.....	146
Figure H.2 – SCT supplied input binding	147
Figure H.3 – ICT supplied input binding (for later binding).....	148
Figure I.1 – Private LN using standardized DOs (Edition 2).....	150
Figure I.2 –Standardized data objects used in standardized LNs.....	151

Figure I.3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B	152
Figure I.4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B.....	153
Figure I.5 – Edition 1 device: Logical nodes in technical reports	154
Figure I.6 – Edition 2 device: Logical nodes in technical reports	155
Figure I.7 – Edition 2.1 device: Logical nodes in technical reports defining new CDCs.....	156
Figure I.8 – Edition 2.1 device: standardized LNs extended by other domains	157
Figure I.9 – Edition 2 device: standardized LN introduced in IEC 61850-7-420:2009	158
Figure I.10 – Edition 2.1 device: standardized LN introduced in IEC 61850-7-420:2015	159
Figure I.11 – Edition 2 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007	160
Figure I.12 – Edition 2.1 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007.....	161
Figure J.1 – Private LN using standardized DOs (Edition 2)	162
Figure J.2 –Standardized data objects used in standardized LNs.....	163
Figure J.3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B	164
Figure J.4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B.....	165
Figure J.5 – Edition 1 device: Logical nodes in technical reports.....	166
Figure J.6 – Edition 2 device: Logical nodes in technical reports.....	167
Figure J.7 – Edition 2.1 device: Logical nodes in technical reports defining new CDCs	168
Figure J.8 – Edition 2.1 device: standardized LNs extended by other domains.....	169
Figure J.9 – Edition 2 device: standardized LN introduced in IEC 61850-7-420:2009	170
Figure J.10 – Edition 2.1 device: standardized LN introduced in IEC 61850-7-420:2015	171
Figure J.11 – Edition 2 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007	172
Figure J.12 – Edition 2.1 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007.....	173
Table 1 – LN groups	21
Table 2 – Logical node class XCBR (conceptual).....	32
Table 3 – Excerpt of integer status setting	40
Table 4 – Comparison of the data access methods	45
Table 5 – ACSI models and services.....	77
Table 6 – Logical node circuit breaker.....	88
Table 7 – Controllable double point (DPC).....	90
Table 8 – ACSI class definition	93
Table 9 – Single point status common data class (SPS).....	95
Table 10 – Quality components attribute definition.....	96
Table 16 – Attributes of DetailQual	97
Table 11– Functional constraints (excerpt).....	98
Table 12– Trigger option.....	98
Table 13 – GenLogicalNodeClass definition.....	100
Table 14 – Excerpt of logical node name plate common data class (LPL).....	114
Table 15 – Excerpt of the abstract common data class BasePrimitiveCDC	114

Table A.2 – List of common data classes (excerpt)	125
Table K.1 – Information users and information providers.....	175
Table K.2 – Data model use cases	177
Table K.3 – Services use cases	179
Table K.4 – Data Forbidden cases	180
Table K.5 – Add new basic type	181
Table K.6 – Extend CDC with DA of new FC	182
Table K.7 – Add new DA using existing type and FCs	182
Table K.8 – Add a new DO based on new CDC using existing types and FCs	183
Table K.9 – Adding a new DO based on existing CDC	184
Table K.10 – Renaming a DO	184
Table K.11 – Renaming a DA, subDO or subDA.....	185
Table K.12 – Deprecation of a FC	186
Table K.13 – Deprecation of a DA.....	186
Table K.14 – Removal of a DA.....	187
Table K.15 – Deprecation of a DO	187
Table K.16 – Use of a weaker presence condition.....	188
Table K.17 – Use of a stronger presence condition	188
Table K.18 – Extend an enumeration with an enumerated value	189
Table K.19 – Modification of an enumerated value from an enumeration list	189
Table K.20 – Deprecate an enumerated value from an enumeration list.....	190
Table K.21 – Extension of a PACKED LIST.....	190
Table K.22 – Extend size of names / references	191
Table K.23 – Add new basic type	191
Table K.24 – Adding new Control Blocks.....	192
Table K.25 – Adding new services	192
Table K.26 – Add new attributes in control block.....	193
Table K.27 – Deprecating control block class.....	193
Table K.28 – Using a stronger presence condition	194

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY AUTOMATION –

Part 7-1: Basic communication structure – Principles and models

FOREWORD

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

This consolidated version of the official IEC Standard and its amendment has been prepared for user convenience.

IEC 61850-7-1 edition 2.1 contains the second edition (2011-07) [documents 57/1121/FDIS and 57/1145/RVD] and its amendment 1 (2020-08) [documents 57/2201/FDIS and 57/2221/RVD].

International Standard IEC 61850-7-1 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

This second edition edition constitutes a technical revision.

Compared to the first edition, this second edition introduces:

- the model for statistical and historical statistical data,
- the concepts of proxies, gateways, LD hierarchy and LN inputs,
- the model for time synchronisation,
- the concepts behind different testing facilities,
- the extended logging function.

It also clarifies the following points:

- the use of numbers for data extension,
- the use of namespaces,
- the mode and behaviour of a logical node,
- the use of range and deadbanded values,
- the access to control actions and others.

Compared to the second edition, this first revision of the second edition:

- provides clarifications and corrections to the second edition of IEC 61850-7-1, based 55 on the tissues = {828, 874, 948, 1060, 1072, 1129, 1151, 1196, 1251, 1268, 1312, 1396, 1437, 1468, 1491};
- re-edit the Clause 8.2.3 on gateways and proxies;
- re-edit the complete Clauses 13 and 14 about namespaces and rules for extension of object classes;
- include three new normative annexes respectively for LGOS/LSVS engineering (G), GOOSE/SMV Subscription Configuration (H) and use case scenarios examples for clarifying the common rules of Clause 14 (J);
- introduces in the informative annex I, the concept of decoupling domain namespaces from basic namespaces in edition 3 of this document;

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

A list of all parts of the IEC 61850 series, under the general title: *Communication networks and systems for power utility automation*, can be found on the IEC website.

The committee has decided that the contents of the base publication and its amendment will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This part of the IEC 61850 series provides an overview of the architecture for communication and interactions between systems for power utility automation such as protection devices, breakers, transformers, substation hosts etc.

This document is part of a set of specifications which details a layered communication architecture for power utility automation. This architecture has been chosen to provide abstract definitions of classes (representing hierarchical information models) and services such that the specifications are independent of specific protocol stacks, implementations, and operating systems.

The goal of the IEC 61850 series is to provide interoperability between the IEDs from different suppliers or, more precisely, between functions to be performed by systems for power utility automation but residing in equipment (physical devices) from different suppliers. Interoperable functions may be those functions that represent interfaces to the process (for example, circuit breakers) or substation automation functions such as protection functions. This part of the IEC 61850 series uses simple examples of functions to describe the concepts and methods applied in the IEC 61850 series.

This part of the IEC 61850 series describes the relationships between other parts of the IEC 61850 series. Finally this part defines how interoperability is reached.

NOTE Interchangeability is the ability to replace a device from the same vendor, or from different vendors, utilising the same communication interface and as a minimum, providing the same functionality, with no impact on the rest of the system. If differences in functionality are accepted, the exchange may also require some changes somewhere else in the system. Interchangeability implies a standardisation of functions and, in a strong sense, of devices which are outside the scope of this standard. Interchangeability is outside the scope, but it will be supported following this standard for interoperability.

This part of the IEC 61850 series is intended for all stakeholders of standardised communication and standardised systems in the utility industry. It provides an overview of and an introduction to IEC 61850-7-4, IEC 61850-7-3, IEC 61850-7-2, IEC 61850-6, and IEC 61850-8-1.

COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY AUTOMATION –

Part 7-1: Basic communication structure – Principles and models

1 Scope

This part of the IEC 61850 series introduces the modelling methods, communication principles, and information models that are used in the various parts of the IEC 61850-7-x series. The purpose of this part of the IEC 61850 series is to provide – from a conceptual point of view – assistance to understand the basic modelling concepts and description methods for:

- substation-specific information models for power utility automation systems,
- device functions used for power utility automation purposes, and
- communication systems to provide interoperability within power utility facilities.

Furthermore, this part of the IEC 61850 series provides explanations and provides detailed requirements relating to the relation between IEC 61850-7-4, IEC 61850-7-3, IEC 61850-7-2 and IEC 61850-5. This part explains how the abstract services and models of the IEC 61850-7-x series are mapped to concrete communication protocols as defined in IEC 61850-8-1.

The concepts and models provided in this part of the IEC 61850 series may also be applied to describe, among others, information models and functions for:

- hydroelectric power plants,
- substation to substation information exchange,
- information exchange for distributed automation,
- substation to control centre information exchange,
- information exchange for metering,
- condition monitoring and diagnosis, and
- information exchange with engineering systems for device configuration.

NOTE 1 This part of IEC 61850 uses examples and excerpts from other parts of the IEC 61850 series. These excerpts are used to explain concepts and methods. These examples and excerpts are informative in this part of IEC 61850.

NOTE 2 Examples in this part use names of classes (e.g. XCBR for a class of a logical node) defined in IEC 61850-7-4, IEC 61850-7-3, and service names defined in IEC 61850-7-2. The normative names are defined in IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2 only.

NOTE 3 This part of IEC 61850 does not provide a comprehensive tutorial. It is recommended that this part be read first – in conjunction with IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2. In addition, it is recommended that IEC 61850-1 and IEC 61850-5 also be read.

NOTE 4 This part of IEC 61850 does not discuss implementation issues.

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

IEC 61850-3, *Communication networks and systems in substations – Part 3: General requirements*

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

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

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

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

IEC 61850-10, *Communication networks and systems in substations – Part 10: Conformance testing*

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO/IEC 8825 (all parts), *Information technology – ASN.1 encoding rules*

ISO 9506-1, *Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*

ISO 9506-2, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

3 Terms and definitions

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

3.1 information

knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning

[IEC 60050-101:1998, 101-12-01]

3.2 information model

knowledge concerning power utility functions and devices in which the functions are implemented

This knowledge is made visible and accessible through the means of the IEC 61850 series. The model describes in an abstract way a communication-oriented representation of a real function or device.

3.3 model

a representation of some aspect of reality

The purpose of creating a model is to help understand, describe, or predict how things work in the real world by exploring a simplified representation of a particular entity or phenomenon. The focus of the model defined in IEC 61850-7-x is on the communication features of the data and functions modelled.

4 Abbreviated terms

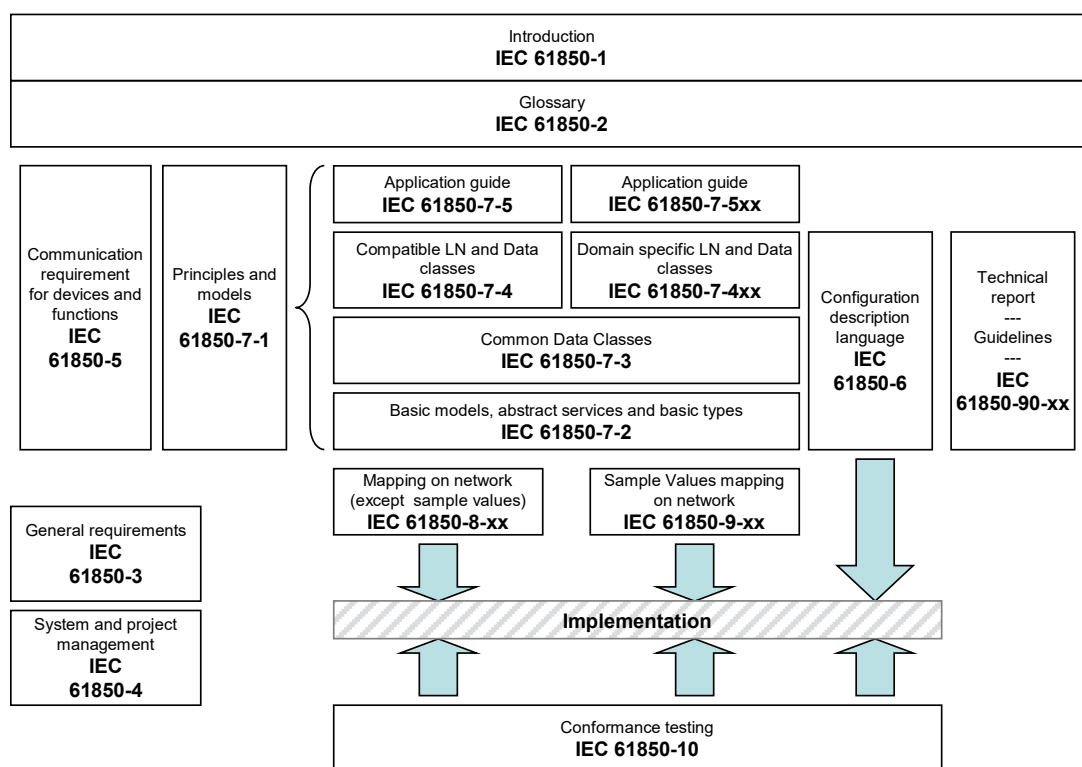
ACSI	Abstract communication service interface
ASN.1	Abstract syntax notation one
API	Application program interface
CDC	Common data class
CT	Current transformer
DST	Daylight saving time
GOOSE	Generic oriented object system event
IED	Intelligent electronic device
LD	Logical device
LN	Logical node
LLN0	Logical node zero
LPHD	Logical node physical device
MICS	Model implementation conformance statement
MMS	Manufacturing message specification
PHD	Physical device
PICOM	Piece of communication
PICS	Protocol implementation conformance statement
SAS	Substation automation system
SCSM	Specific communication service mapping
SICS	SCL implementation conformance statement
SoE	Sequence of events
SMV	Sample values
UCAIug	UCA international users group
UTC	Universal time coordinated
VMD	Virtual manufacturing device

VT Voltage transformer
 XML extended markup language

5 Overview of the IEC 61850 series concepts

5.1 Objective

IEC 61850-7-4, IEC 61850-7-3, IEC 61850-7-2, IEC 61850-6, and IEC 61850-8-1 are closely related. This subclause provides an overview of these parts and it describes how they are interwoven. The modelling and implementation methods applied in the different parts of the standard and their relation are shown in Figure 1.



IEC 1402/11

Figure 1 – Relations between modelling and mapping parts of the IEC 61850 series

Each part defines a specific aspect of a substation IED:

- IEC 61850-1 gives an introduction and overview of the IEC 61850 series,
- IEC 61850-2 contains the glossary of specific terminology and definitions used in the context of power utility automation systems within the various parts of the standard,
- IEC 61850-3 specifies the general requirements of the communication network with regard to the quality requirements, environmental conditions and auxiliary services,
- IEC 61850-4 pertains to the system and project management with respect to the engineering process, the life cycle of the SAS and the quality assurance from the development stage to the discontinuation and decommissioning of the SAS.

- IEC 61850-5 specifies the communication requirements of the functions being performed in systems for power utility automation and to device models. All known functions and their communication requirements are identified,
- this part of IEC 61850 defines the basic principles and modelling methods,
- IEC 61850-6 specifies a file format for describing communication related IED (intelligent electronic device) configurations and IED parameters, communication system configurations, switchyard (function) structures, and the relations between them. The main purpose of the format is to exchange IED capability descriptions, and system level descriptions between engineering tools of different manufacturers in a compatible way. The defined language is called substation configuration description language (SCL). Mapping specific extensions or usage rules may be required in the appropriate parts.
- IEC 61850-7-5 defines the usage of information models for substation automation applications. It gives clear examples on how to apply LNs and data defined in IEC 61850-7-4 for different substation applications. The examples cover applications from monitoring function to protection blocking schemes. Other domain specific application guides which are within the scope of IEC technical committee 57 are defined in the IEC 61850-7-5xx series¹. Examples are hydropower and distributed energy resources domains,
- IEC 61850-7-4 defines specific information models for substation automation functions (for example, breaker with status of breaker position, settings for a protection function, etc.) – what is modelled and could be exchanged. Other domain specific information models within the scope of IEC technical committee 57 are defined in the 61850-7-4xx series,
- IEC 61850-7-3 has a list of commonly used information (for example, for double point control, 3-phase measurand value, etc.) – what the common basic information is,
- IEC 61850-7-2 provides the services to exchange information for the different kinds of functions (for example, control, report, get and set, etc.) – how to exchange information,
- IEC 61850-8-1 defines the concrete means to communicate the information between IEDs (for example, the application layer, the encoding, etc.) – how to serialise the information during the exchange,
- IEC 61850-9-2, and particularly the subset 9-2LE described in the “Implementation Guideline for Digital Interface to Instrument Transformers using IEC 61850-9-2” by the UCAlug, defines the concrete means to communicate sampled values between sensors and IEDs,
- IEC 61850-10 defines the methods and abstract cases for conformance testing of devices and engineering tools as well as the metrics to be measured within devices according to the requirements defined in IEC 61850-5,
- there may be object classes defined for various other application domains outside the scope of IEC technical committee 57. They are relevant to Figure 1 only if they are built according to the approach of the IEC 61850 series.

5.2 Topology and communication functions of substation automation systems

As shown by the topology in Figure 2, one focus of the IEC 61850 series is the support of substation automation functions by the communication of (numbers in brackets refer to the figure):

- sampled value exchange for CTs and VTs (1),
- fast exchange of I/O data for protection and control (2),
- control signals (3),
- trip signals (4),

¹ IEC 61850-7-4xx, -7-5xx, -8-xx, -9-xx and -90-xx are series of documents whose scope is similar. For example, IEC 61850-7-4 deals with data object classes used for substations while IEC 61850-7-410 deals with data object classes used for hydroelectric power plants. IEC 61850-90-xx series is reserved for technical reports or guidelines.

- engineering and configuration (5),
- monitoring and supervision (6),
- control-center communication (7),
- time-synchronisation,
- etc.

Support for other functions such as metering, condition monitoring, and asset management is provided as well.

Many functions are implemented in intelligent electronic devices (IED). Several functions may be implemented in a single IED or one function may be implemented in one IED and another function may be hosted by another IED. IEDs (i.e., the functions residing in IEDs) communicate with functions in other IEDs by the information exchange mechanisms of this standard. Therefore, functions distributed over more than one IED may be also implemented.

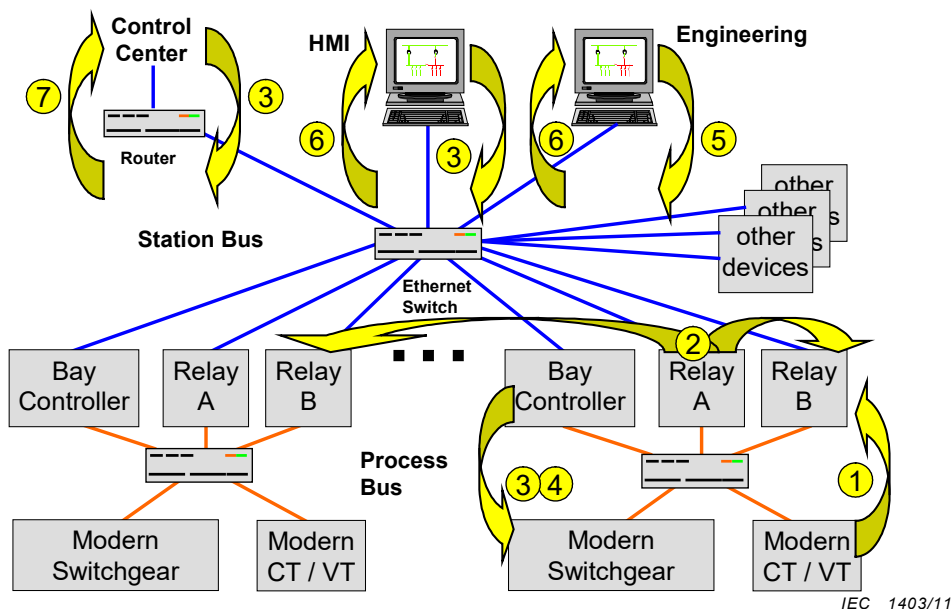


Figure 2 – Sample substation automation topology

5.3 The information models of substation automation systems

The information exchange mechanisms rely primarily on well-defined information models. These information models and the modelling methods are at the core of the IEC 61850 series. The IEC 61850 series uses the approach to model the common information found in real devices as depicted in Figure 3. All information made available to be exchanged with other devices is defined in the standard. The model provides for systems for power utility automation an image of the analogue world (power system process, switchgear).

NOTE 1 “The common information” in the context of the IEC 61850 series means that the stakeholders of power utility automation systems (users and vendors) have agreed that the information defined in the IEC 61850 series is widely accepted and required for the open exchange of information between any kind of substation IEDs.

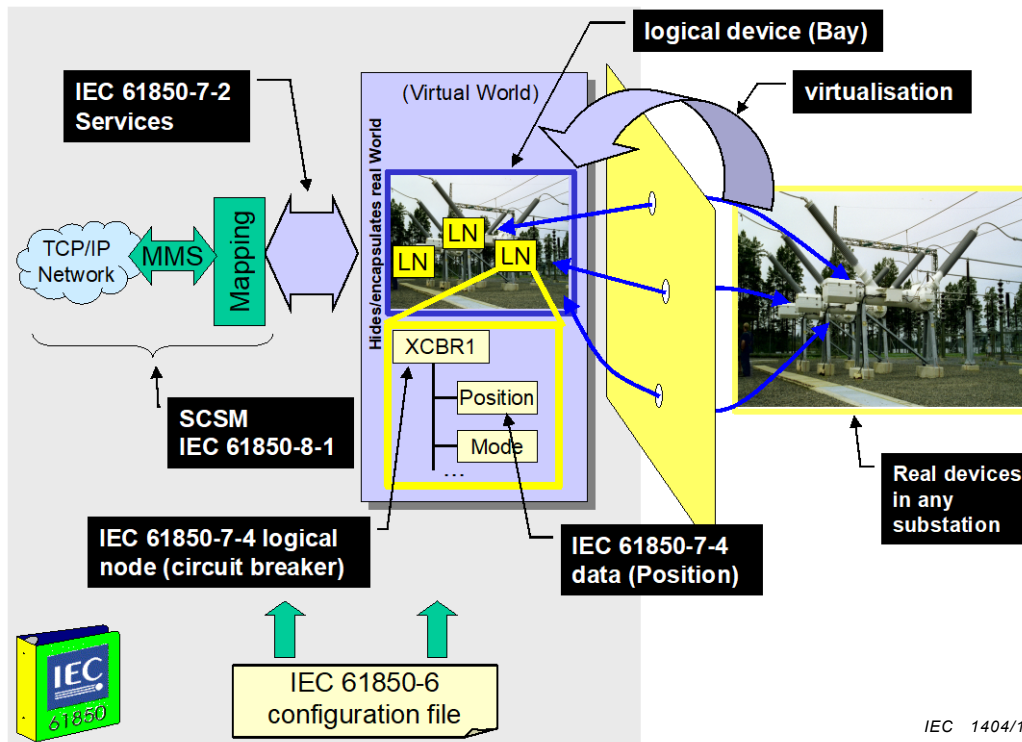


Figure 3 – Modelling approach (conceptual)

Implementations to reach interoperability have to be based on common understanding of definitions. Therefore, the parts describing the data model contain mandatory semantic tables which have to be considered very carefully.

The IEC 61850 series defines the information and information exchange in a way that it is independent of a concrete implementation (i.e., it uses abstract models). The standard also uses the concept of virtualisation. Virtualisation provides a view of those aspects of a real device that are of interest for the information exchange with other devices. Only those details that are required to provide interoperability of devices are defined in the IEC 61850 series.

As described in IEC 61850-5, the approach of the standard is to decompose the application functions into the smallest entities, which are used to exchange information. The granularity is given by a reasonable distributed allocation of these entities to dedicated devices (IED). These entities are called logical nodes (for example, a virtual representation of a circuit breaker class, with the standardised class name XCBR). The logical nodes are modelled and defined from the conceptual application point of view in IEC 61850-5. Several logical nodes build a logical device (for example, a representation of a Bay unit). A logical device is always implemented in one IED; therefore logical devices do not contain logical nodes from different IEDs².

Real devices on the right-hand side of Figure 3 are modelled as a virtual model in the middle of the figure. The logical nodes defined in the logical device (for example, bay) correspond to well-known functions in the real devices. In this example, the logical node XCBR represents a specific circuit breaker of the bay to the right.

NOTE 2 The logical nodes of this example may be implemented in one or several IEDs as appropriate. If the logical nodes are implemented in different IEDs, they need exchange information over a network. Information exchange inside a logical node is outside the scope of the IEC 61850 series.

² There is one exception: the proxy/gateway application. For details, refer to 8.2.3 Gateways and proxies.

Based on their functionality, a logical node contains a list of data (for example, position) with dedicated data attributes. The data have a structure and a well-defined semantic (meaning in the context of systems for power utility automation or, e.g. more specifically, of substation automation systems). The information represented by the data and their attributes are exchanged by the services according to the well-defined rules and the requested performance as described in IEC 61850-5. The services are implemented by a specific and concrete communication means (SCSM, for example, using MMS, TCP/IP, and Ethernet among others).

The logical nodes and the data contained in the logical device are crucial for the description and information exchange for substation automation systems to reach interoperability.

The logical devices, the logical nodes and the data they contain need to be configured. The main reason for the configuration is to select the appropriate logical nodes and data from the standard and to assign the instance-specific values, for example, concrete references between instances of the logical nodes (their data) and the exchange mechanisms, and initial values for process data.

5.4 Applications modelled by logical nodes defined in IEC 61850-7-4

Table 1 lists all groups of logical nodes defined in IEC 61850-7-4. Over one hundred logical nodes covering the most common applications of substation and feeder equipment are defined. While the definition of information models for protection and protection related applications is important because of the high impact of protection for safe and reliable operation of the power system, the covered applications include many other functions like monitoring, measurement, control and power quality.

Table 1 – LN groups

Group indicator	Logical node groups
A	Automatic control
C	Supervisory control
D	DER (Distributed Energy Resources)
F	Functional blocks
G	Generic function references
H	Hydro power
I	Interfacing and archiving
K	Mechanical and non-electrical primary equipment
L	System functions
M	Metering and measurement
P	Protection functions
Q	Power quality events detection related
R	Protection related functions
S	Supervision and monitoring
T	Instrument transformer and sensors
W	Wind power
X	Switchgear
Y	Power transformer and related functions
Z	Further (power system) equipment

IEC 61850 has well-defined rules to define additional logical nodes and data, for example for additional functions within substations or for other application domains such as wind power plants. For details on the extension rules, see Clauses 13 and 14 of this document.

The following excerpt of the logical nodes has been included to provide an example of what kind of real applications the logical nodes represent:

- distance protection;
- differential protection;
- overcurrent;
- undervoltage;
- directional over power;
- volts per Hz relay;
- transient earth fault;
- directional element;
- harmonic restraint;
- protection scheme;
- zero speed or underspeed;
- measurement;
- metering;
- sequence and imbalance;
- harmonics and interharmonics;
- differential measurements;
- switch control;
- circuit breaker;
- circuit switch;
- and others.

Most logical nodes provide information that can be categorised as depicted in Figure 4. The semantic of a logical node is represented by data and data attributes. Logical nodes may provide a few or up to 30 data. Data may contain a few or even more than 20 data attributes. Logical nodes may contain more than 100 individual information (points) organised in a hierarchical structure.

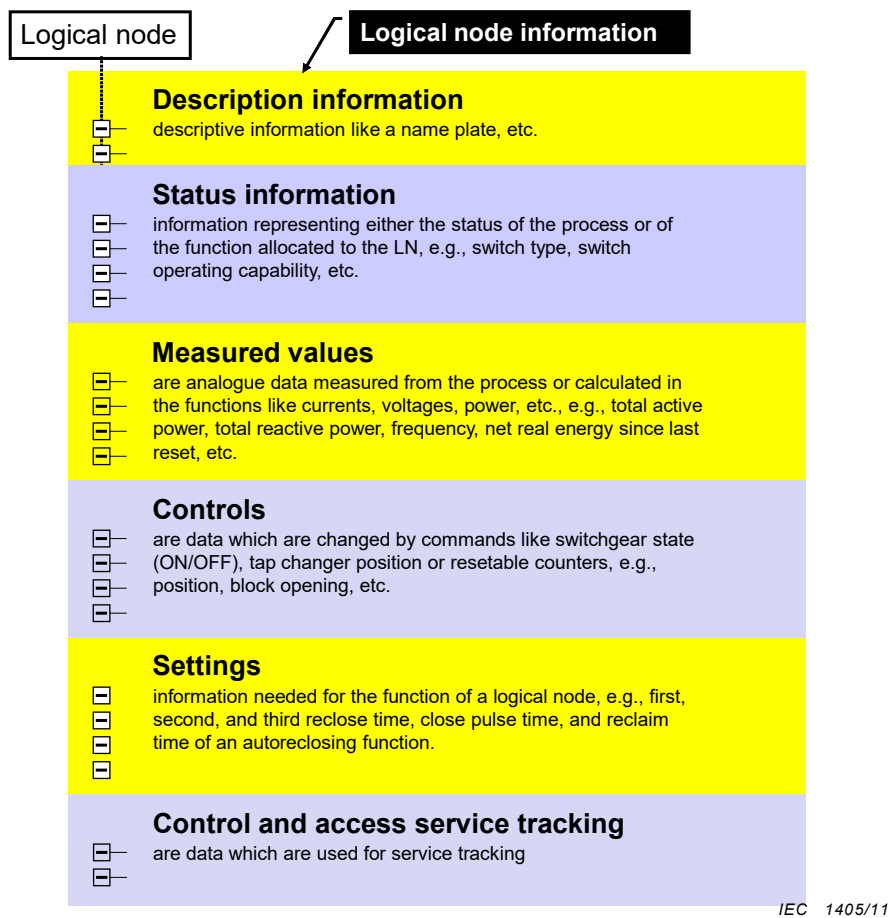


Figure 4 – Logical node information categories

IEDs are built up by composing logical nodes as depicted in Figure 5. The logical nodes are the building blocks of substation IEDs, for example, circuit breaker (XCBR) and others. In the example for each phase, one instance of XCBR is used.

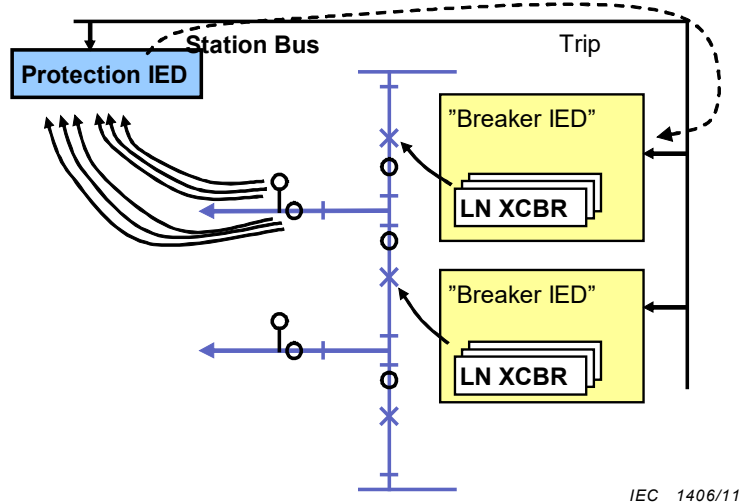


Figure 5 – Build-up of devices (principle)

In Figure 5, the protection IED receives the values for the voltage and current from conventional VT and CT. The protection functions in the protection device may detect a fault and issue or send a trip signal via the station bus. The standard supports also IEDs for digitizing VTs and CTs sending voltage and current as samples to the protection over a serial link. The output of conventional VTs and CTs may also be converted at the source to samples and transmitted over this serial link.

The logical nodes are used to build up substation IEDs.

5.5 The semantic is attached to data

The mean number of specific data provided by logical nodes defined in IEC 61850-7-4 is approximately 20. Each of the data (for example, position of a circuit breaker) comprises several details (the data attributes). The position (named “Pos”) of a circuit breaker is defined in the logical node XCBR (see Figure 6). The position is defined as data. The category of the position in the logical node is “controls” – the position can be controlled via a control service.

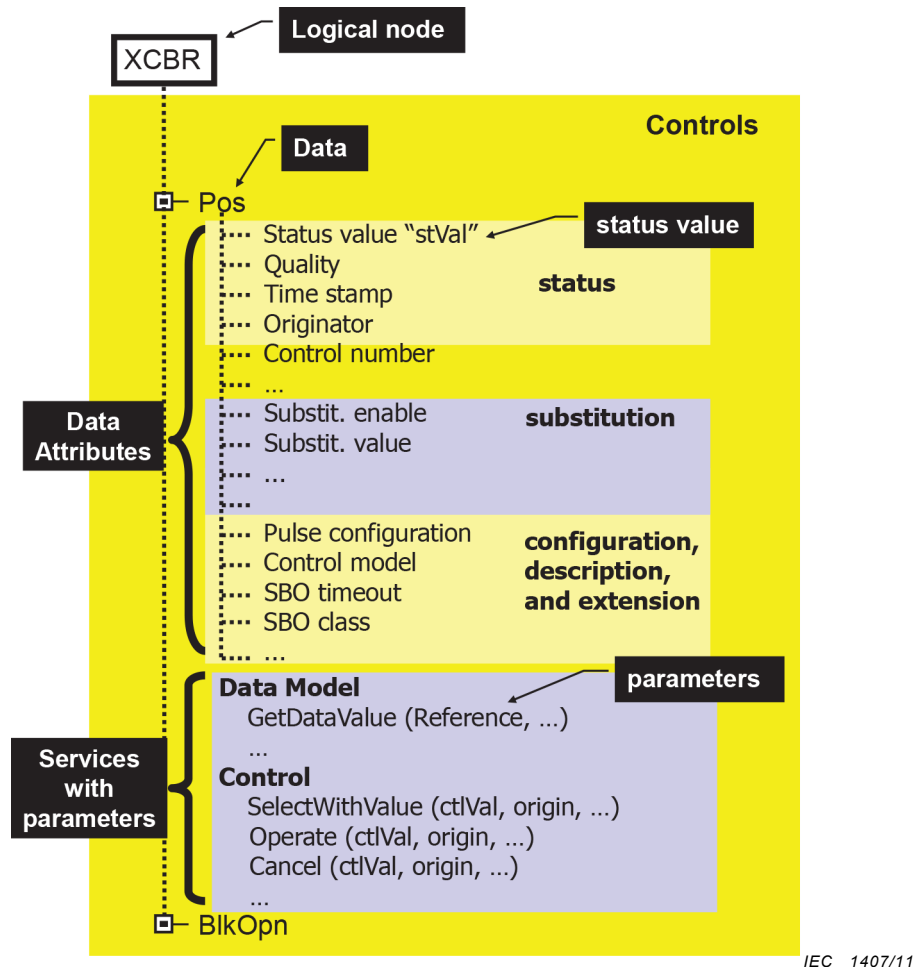


Figure 6 – Position information depicted as a tree (conceptual)

The position Pos is more than just a simple “point” in the sense of simple RTU protocols. It is made up of several data attributes. The data attributes are categorised as follows:

- status (or measured/metered values, or settings),
- control mirror,
- substitution and blocked,
- configuration, description and extension.

The data example Pos has approximately 20 data attributes accessible through different services. The data attribute Pos.stVal represents the position of the real breaker (could be in intermediate-state, off, on, or bad-state).

The position Pos can be controlled by use of control services and the associated service parameters. It is important to understand that these service parameters are not part of the data model: they do not represent data attributes. They only “live” the time of the command execution.

The position also has information about the originator that issued the command and the control number (given by the originator in the request). Furthermore, the position contains the cause diagnosis of a negative control response. The quality and time stamp information indicate the current validity of the status value and the time of the last change of the status value.

The current values for stVal, the quality and the time stamp (associated with the stVal) can be read, reported or logged in a buffer of the IED.

The values for stVal and quality can be remotely substituted. The substituted values take effect immediately after enabling substitution.

Several data attributes are defined for the configuration of the control behaviour, for example, pulse configuration (single pulse or persistent pulses, on/off-duration, and number of pulses) or control model (direct, select-before-operate, etc.).

Data attributes are defined primarily by an attribute name and an attribute type:

Attribute name	Attribute type	FC	TrgOp	(Value/value range) Description	PresCond
stVal	DpStatusKind	ST	dchg	Status value of the controllable data object.	M
ctlModel	CtlModelKind	CF	dchg	Control model as defined in IEC 61850-7-2 that reflects the behaviour of the controllable data object.	M

Additional information provides further details (one could say provides meta-data) on:

- the services allowed: functional constraint -> FC=SV means that specific services shall be applied only (for example SV refers to the substitution service),
- the trigger conditions that cause a report to be sent: TrgOp=dchg means that a change in the value of that attribute causes a report,
- the value or value range,
- the presence conditions, e.g. mandatory (M) or optional (O). These conditions specify presence of elements in a given context (one LN, or one CDC, or one data attribute type, or one data object for dataNs) and are specified in IEC 61850-7-2.

The data attribute names are standardised (i.e., they are reserved) names that have a specific semantic in the context of the IEC 61850 series. The semantic of all data attribute names is defined in IEC 61850-7-3.

The names of the data and data attributes carry the crucial semantic of a substation IED.

The position information Pos as shown in Figure 6 has many data attributes that can found in many other switching-specific applications. The prime characteristic of the position is the data attribute stVal (status value) which represents four states: intermediate-state | off | on | bad-state. These four states (represented usually with two bits) are commonly known as “double point” information. The whole set of all the data attributes defined for the data Pos (position)

is called a “common data class” (CDC). The name of the common data class of the double point information is DPC (controllable double point).

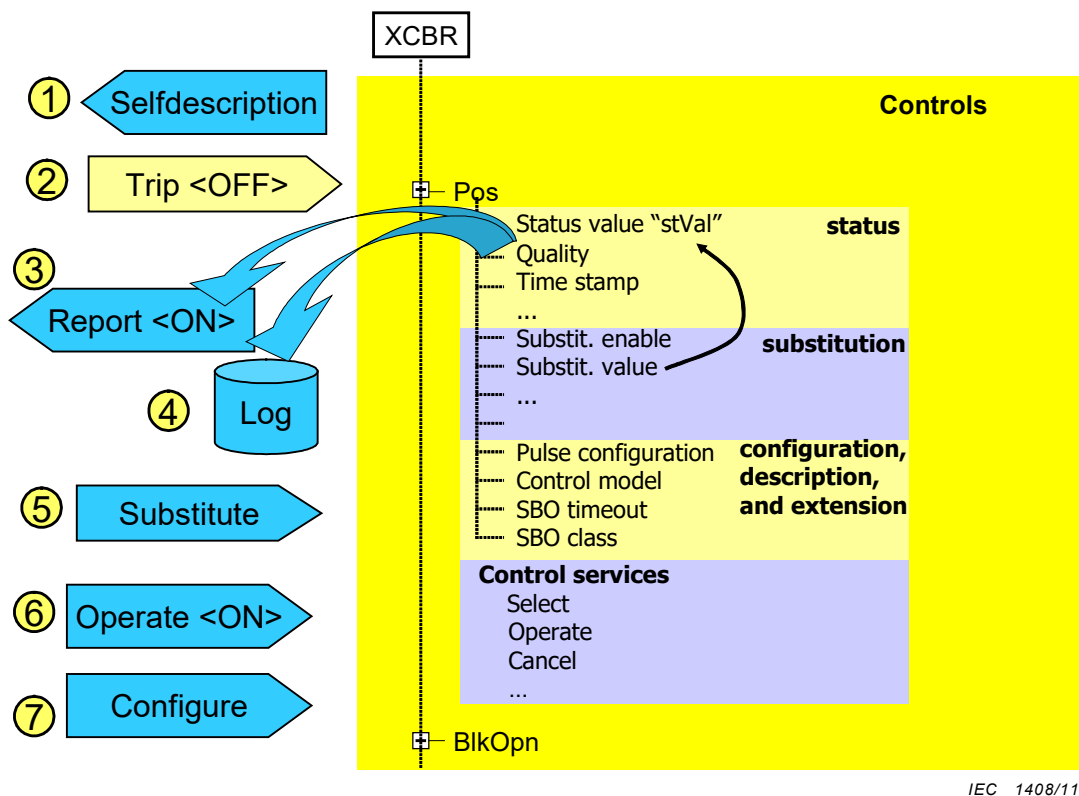
Common data classes provide a useful means to reduce the size of data definitions (in the standard). The data definition does not need to list all the attributes but needs to just reference the common data class. Common data classes are also very useful to keep the definitions of data attributes consistent. A change in the double point control CDC specific data attributes only needs to be made in a single place – in the DPC definition of IEC 61850-7-3.

IEC 61850-7-3 defines common data classes for a wide range of well-known applications. The core common data classes are classified into the following groups:

- status information,
- measurand information,
- controllable status information,
- controllable analogue information,
- status settings,
- analogue settings,
- description information, and
- Service tracking.

5.6 The services to exchange information

The logical nodes, data, data attributes and service parameters are defined mainly to specify the information required to perform an application, and for the exchange of information between IEDs. The information exchange is defined by means of services. An excerpt of the services is displayed in Figure 7.



NOTE The circles with the numbers (1) to (7) refer to the list below.

Figure 7 – Service excerpt

The operate service manipulates the control specific service parameters of a circuit breaker position (open or close the breaker). The report services inform another device that the position of the circuit breaker has been changed. The substitute service forces a specific data attribute to be set to a value independent of the process.

The categories of services (defined in IEC 61850-7-2) are as follows:

- retrieving the self-description of a device, see (1) in Figure 7,
- fast and reliable peer-to-peer exchange of status information (tripping or blocking of functions or devices), see (2) in Figure 7,
- reporting of any set of data (data attributes), SoE – cyclic and event triggered, see (3) in Figure 7,
- logging and retrieving of any set of data (data attributes) – cyclic and event, see (4) in Figure 7,
- substitution, see (5) in Figure 7,
- handling and setting of parameter setting groups,
- transmission of sampled values from sensors,
- time synchronisation,
- file transfer,
- control devices (operate service), see (6) in Figure 7, and
- online configuration, see (7) in Figure 7.

Many services operate directly on the attributes of the information model (i.e., on the data attributes of data contained in logical nodes). The pulse configuration of the data attribute Pos of a specific circuit breaker can be set directly by a client to a new value. Directly means that the service operates on the request of the client without specific constraints of the IED.

Other services provide a more complex behaviour which is dependent on the state of some specific state machine. A control request may be required to follow a state machine associated with the data attribute, for example, select-before-operate.

There are also several application-specific communication services that provide a comprehensive behaviour model which partially act autonomously. The reporting service model describes an operating-sequence in which the IED acts automatically on certain trigger conditions defined in the information model (for example, report on data-change of a status value) or conditions defined in the reporting service model (for example, report on a periodical event).

5.7 Services mapped to concrete communication protocols

The services defined in IEC 61850-7-2 are called abstract services. Abstract means that only those aspects that are required to describe the required actions on the receiving and sending side of a service request are defined in IEC 61850-7-2. They are based on the functional requirements in IEC 61850-5. The semantic of the service models with their attributes and the semantic of the services that operate on these attributes (including the parameters that are carried with the service requests and responses) are defined in IEC 61850-7-2.

The specific syntax (format) and especially the encoding of the messages that carry the service parameters of a service and how these are passed through a network are defined in a specific communication service mapping (SCSM). One SCSM – IEC 61850-8-1 – is the mapping of the services to MMS (ISO 9506-1 and ISO 9506-2) and other provisions like TCP/IP and Ethernet (see Figure 8), another is IEC 61850-9-2 – the direct mapping on Ethernet.

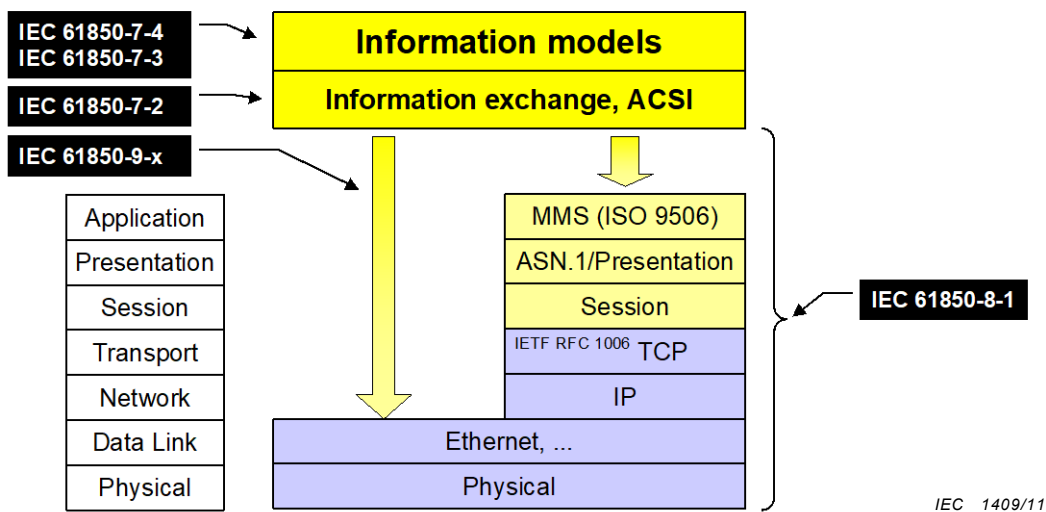


Figure 8 – Example of communication mapping

Additional mappings to other communication stacks are possible. However, to not jeopardize interoperability, the number of accepted mappings in the standard shall be a minimum. The main purpose of this flexibility is to be able to follow over time the evolution in communication technology. The ACSI is independent of the mappings.

5.8 The configuration of the automation system

IEC 61850-6 specifies a file format for describing communication related IED configurations and IED parameters, communication system configurations, switchyard (function) structures, and the relations between them. The main purpose of this format is to exchange IED capability descriptions, and SA system descriptions between IED engineering tools and the system engineering tool(s) of different manufacturers in a compatible way.

The defined language is called substation configuration description language (SCL). The configuration language is based on the extensible markup language (XML) version 1.0.

To support the intended engineering process, the SCL is capable to describe:

- a) a system specification in terms of the single line diagram, and allocation of logical nodes to parts and equipment of the single line to indicate the needed functionality,
- b) pre-configured IEDs with:
 - the logical node, datasets and report control block definitions,
 - the supported services: GOOSE, sampled values, logging, file handling,
- c) pre-configured IEDs with no semantic or a pre-configured semantic for a process part of a certain structure, for example a double busbar GIS line feeder,
- d) complete process configuration with all IEDs bound to individual process functions and primary equipment, enhanced by the access point connections and possible access paths in subnetworks for all possible clients,
- e) as item d) above, but additionally with all predefined associations and client server connections between logical nodes on data level. This is needed if an IED is not capable of dynamically building associations or reporting connections (either on the client or on the server side).

The scope of SCL is focussed on these purposes:

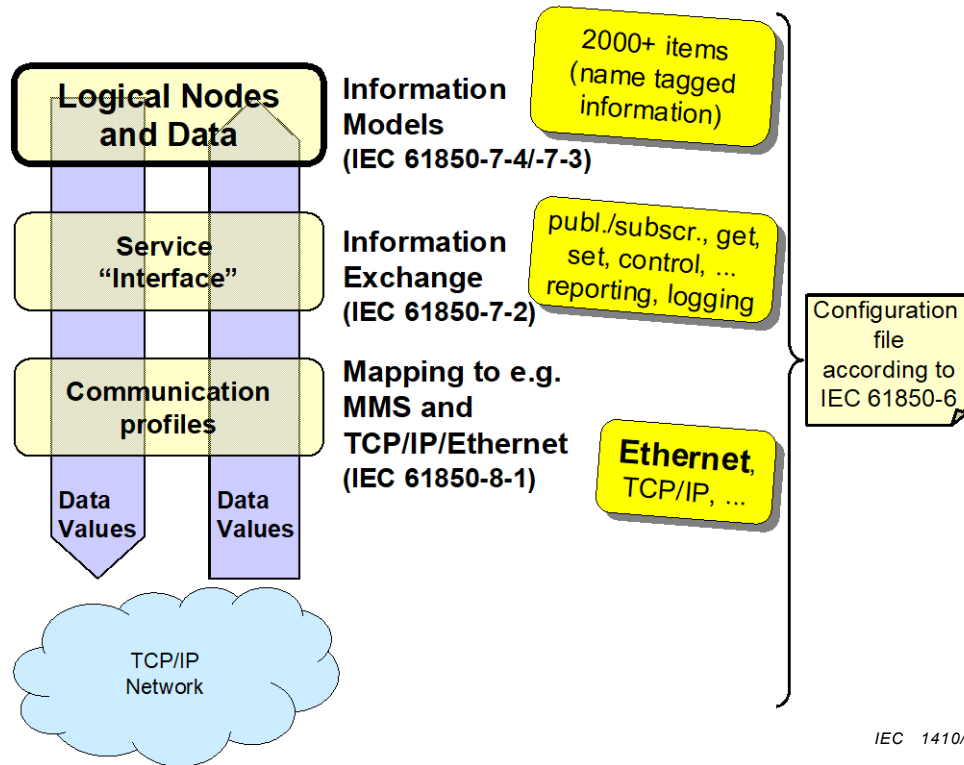
- 1) SAS functional specification (point a) above),
- 2) IED capability description (points b) and c) above), and
- 3) SA system description (points d) and e) above).

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

5.9 Summary

Figure 9 exhibits a summary of Clause 5. The four main building blocks are

- the substation automation system specific information models,
- the information exchange methods,
- the mapping to concrete communication protocols, and
- the configuration of a substation IED.



IEC 1410/11

Figure 9 – Summary

These four building blocks are to a high degree independent of each other. The information is separated from the presentation and from the information exchange services. The information exchange services are separated from the concrete communication profiles. It means that the information models can easily be extended, by definition of new logical nodes and new data according to specific and flexible rules, as required by another application domain. In the same way, different communication stacks may be used following the state-of-the-art in communication technology. But to keep interoperability simple, one stack only should be selected at one time. For the selection, see IEC 61850-8-x and IEC 61850-9-x.

Clause 6 provides a more detailed view of the four building blocks.

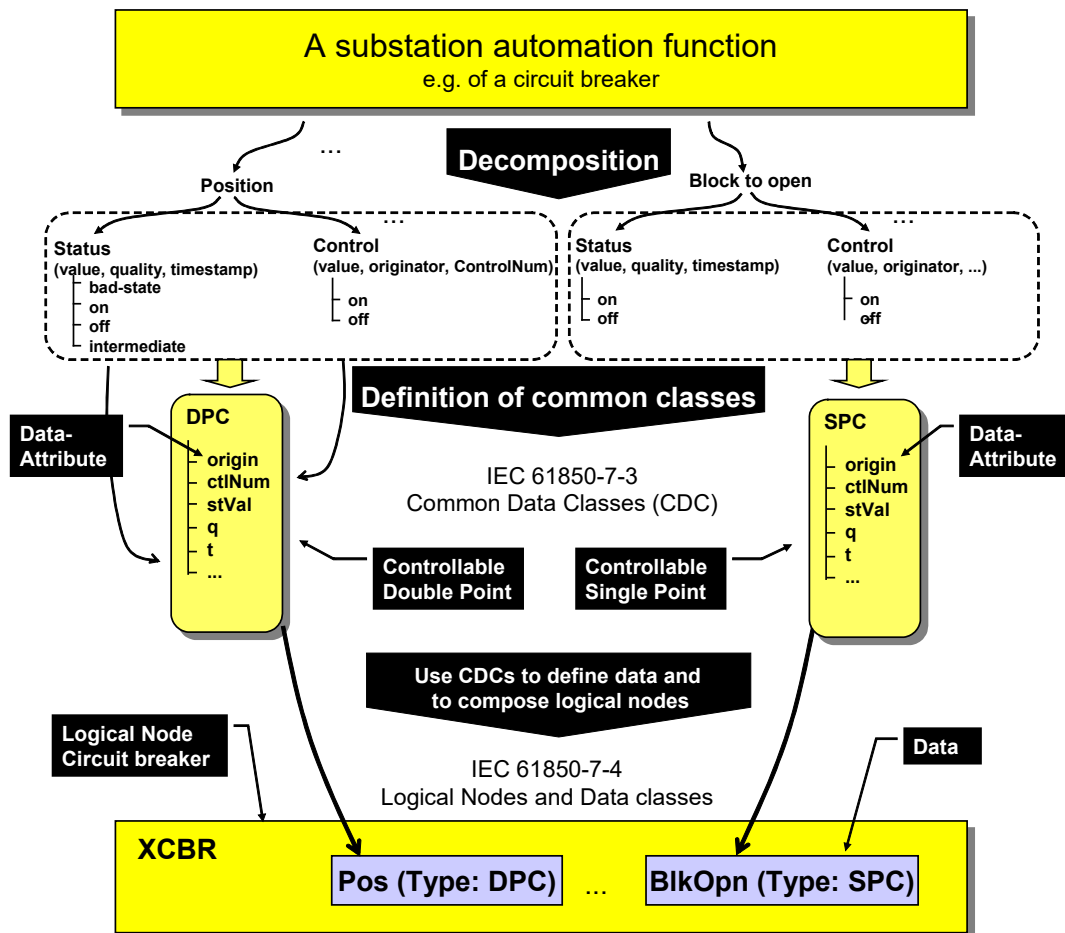
6 Modelling approach of the IEC 61850 series

6.1 Decomposition of application functions and information

As described in IEC 61850-5, the general approach of the IEC 61850 series is to decompose application functions into the smallest entities, which are then used to communicate. The granularity is given by a reasonable distributed allocation of these entities to dedicated devices (IED). The entities are called logical nodes. The requirements for logical nodes are defined – from an application point of view – in IEC 61850-5.

Based on their functionality, these logical nodes comprise data with dedicated data attributes. The information represented by the data and the data attributes are exchanged by dedicated services according to well-defined rules and the performance requested as required in IEC 61850-5.

The decomposition process (to get the most common logical nodes) and the composition process (to build up devices using logical nodes) are depicted in Figure 10. The data classes contained in logical nodes have been defined to support the most common applications in an understandable and commonly accepted way.



IEC 1411/11

Figure 10 – Decomposition and composition process (conceptual)

A small part of a function (an excerpt of a circuit breaker model) has been selected as an example to explain the decomposition process. The circuit breaker has, among many other attributes, a position which can be controlled and monitored and the capability to prevent the switch being opened (for example, by an operator in service situations, block to open). The position comprises some information that represents the status of the position providing the value of the status (on, off, intermediate, bad state), the quality of the value (good, etc.), and the timestamp of the time of the last change of the position. In addition, the function provides the capability to control the switch: Control value (on, off). To keep track of who controlled the switch, the originator stores the information about the entity that issued the last control command. A control number stores the sequence number of the last control command.

The information grouped under the position (status, etc.) represents a very common group of a four-state value that can be reused many times. Similarly, the “block to open” groups information of a two-state value. These groups are called common data classes (CDC):

- four-state reusable class is defined as controllable double point (DPC), and
- two-state reusable class is defined as controllable single point (SPC).

IEC 61850-7-3 defines many common data classes for status, measurands, controllable status, controllable analogue, status settings, and analogue settings.

6.2 Creating information models by stepwise composition

IEC 61850-7-5xx series, IEC 61850-7-4xx series, IEC 61850-7-3, and IEC 61850-7-2 define how to model the information and communication in power utility applications according to the requirements defined in IEC 61850-5. The modelling uses the logical nodes (and their data that represent a huge amount of semantical definitions) primarily as building blocks to compose the visible information of a power utility automation system. The models are used for description of the information produced and consumed by applications and for the exchange of information with other IEDs.

The logical nodes and data classes introduced in IEC 61850-5 are refined and precisely defined in the IEC 61850-7-4xx series. They have been defined in a joint effort of domain experts of the various power utility application domains and modelling experts. The logical nodes and their data are defined with regard to content (semantic) and form (syntax). The approach uses object oriented methods.

NOTE 1 The logical node classes and data classes modelled and defined in IEC 61850-7-4 meet the requirements listed in IEC 61850-5.

In the next step, the common data classes are used to define the (power utility domain-specific) data classes, see lower half of Figure 10. These data classes (defined in IEC 61850-7-4) are specialised common data classes, for example, the data class Pos (a specialisation of DPC) inherits all data attributes of the corresponding common data class DPC, i.e., the stVal, q, t, etc. The semantic of the class Pos is defined at the end of IEC 61850-7-4.

A logical node groups several data classes to build up a specific functionality. The logical node XCBR represents the common information of a real circuit breaker. The XCBR can be reused to describe the common information of circuit breakers of various makes and types.

IEC 61850-7-4 defines several tens of logical nodes making use of hundreds of data names. The logical node XCBR comprises over 15 data classes. An incomplete description of the logical node XCBR is given in Table 2.

Table 2 – Logical node class XCBR (conceptual)

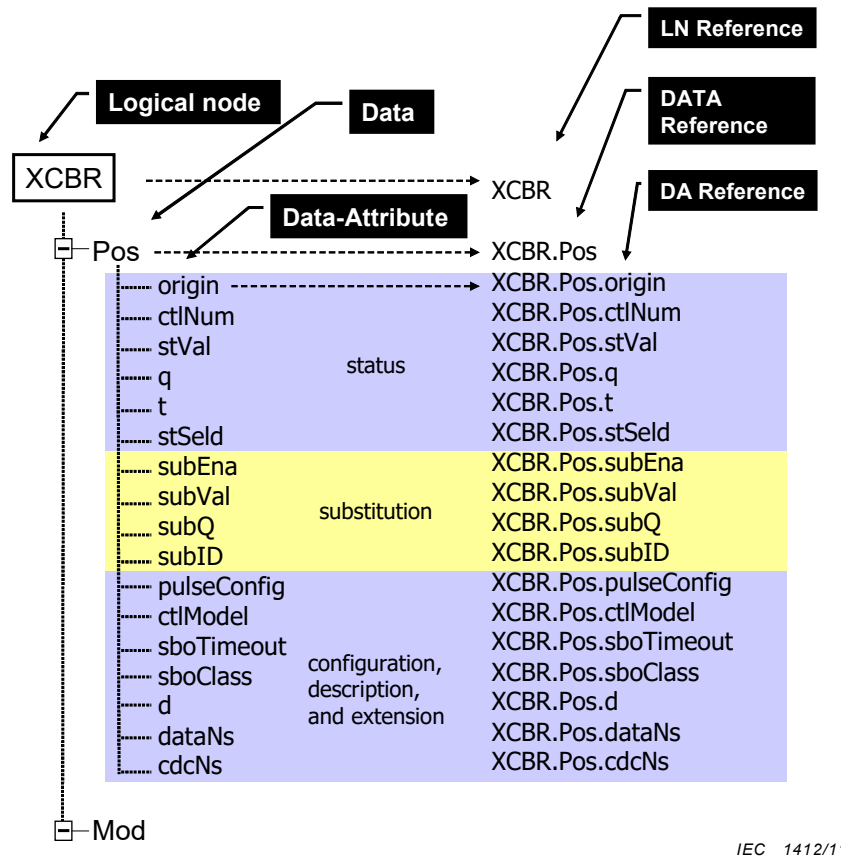
<i>Descriptions</i>
External equipment name plate
Name plate of the logical node
<i>Controls</i>
Switch position (see below for details)
Block opening
Block closing
Charger motor enabled
Control authority at station level. Switches between station and higher level.
...
<i>Status information</i>
Mode
Behaviour
Health
External equipment health
Operation counter
Local operation (Indicates the switchover between local and remote operation; local = TRUE, remote =FALSE)
Circuit breaker operating capability
Point on wave switching capability
Circuit breaker operating capability when fully charged
...

NOTE 2 IEC 61850-7-4 defines a standardised name for each item such as Pos for the switch position. Additionally, the tables for logical nodes contain the common data class to be used for the corresponding data class. Finally, the tables define if the data class in the table is mandatory or optional. These details are explained later in this part.

The content of the marked “switch position” (name = Pos) is introduced in Figure 11.

IEC 61850-7-x series use tables for the definition of the logical node classes and data classes (see IEC 61850-7-4), the common data classes (see IEC 61850-7-3) and service models (see IEC 61850-7-2). Data classes and data attributes form a hierarchical structure as depicted in Figure 11. The data attributes of the data class Pos are functionally grouped (status, substitution, configuration, etc.).

The data attributes have a standardized name and a standardized type. On the right-hand side the corresponding references (object reference) are shown. These references are used to provide the path information to identify the information in the tree.



IEC 1412/11

Figure 11 – XCBR1 information depicted as a tree

XCBR is the root at the level of logical nodes. The object reference XCBR references the complete tree below. XCBR contains data, for example, Pos and Mode. The data Pos (position) is precisely defined in IEC 61850-7-4 (see excerpt of the description):

Description of data

Data object name	Explanation
...	...
Pos	(controllable) Circuit breaker/switch position.
...	...

The content of the position Pos is a list of some 20 data attributes and 7 controllable parameters. The attributes are derived from the common data class DPC (double point control). The data attributes defined in the DPC are partly mandatory and others are optional. Only those data attributes that are required for a specific application are inherited by a data object. For example, if the position does not require the support of substitution, then the data attributes subEna, subVal, subQ, and subID are not required in the data object Pos.

The information exchange services that access the data attributes make use of the hierarchical tree. The data attribute XCBR.PosctlModel defines the type of control service which is supported. The status information could be referenced as a member (XCBR.Pos.stVal) of a data set named "AlarmXCBR". The data set could be referenced by a reporting control block named "Alarm". The report control block could be configured to send a report to a specific computer each time a circuit breaker changes its state (from open to close or from close to open).

6.3 Example of an IED composition

Figure 12 shows examples of different logical nodes being parts of IEDs. The logical nodes involved are PTOC (time overcurrent protection), PDIS (distance protection), PTRC (trip conditioning) and XCBR (circuit breaker). Case 1 shows a protection device with two functions, which are hardwired with the circuit breaker. Case 2 shows a protection device with two functions where the trip is communicated via a trip message over a network to the circuit breaker LN. Case 3 shows the two protection functions in dedicated devices, which may operate both in a fault and where the trips are transmitted as trip messages via the network independently to the circuit breaker LN (XCBR).

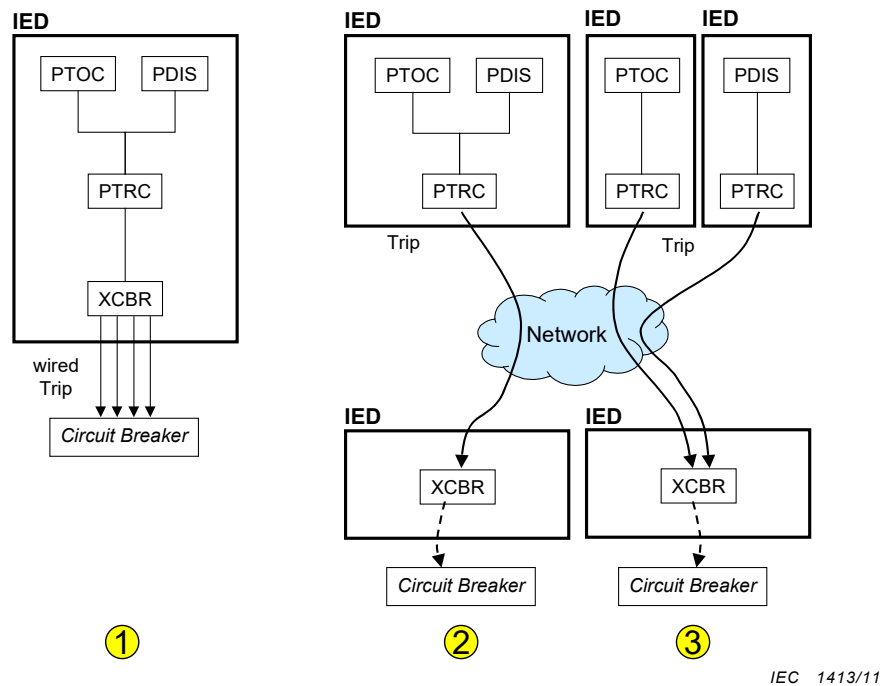


Figure 12 – Example of IED composition

In cases 2 and 3, the IED that hosts the XCBR LNs may be integrated in the real circuit breaker device or hardwired with it as in case 1, but this is outside the scope of the IEC 61850 series. The real breaker is represented for the substation automation system according to the IEC 61850 series by the XCBR LNs.

The IED composition is very flexible to meet current and future needs.

6.4 Information exchange models

6.4.1 General

The information contained in the hierarchical models of IEC 61850-7-4 can be communicated using services defined in IEC 61850-7-2. The information exchange methods (depicted in Figure 12) fall mainly into three categories:

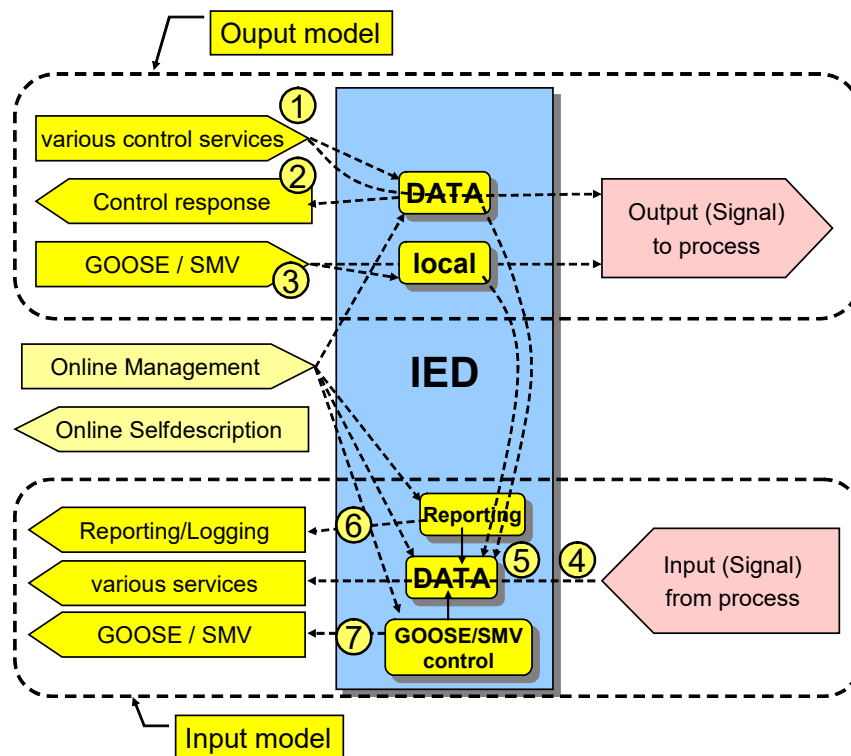
- the output model,
- the input model, and
- the model for online management and self-description.

Several services are defined for each model. The services operate on data, data attributes, and other attributes usually contained in logical nodes.

NOTE 1 Services operate actually on instances of data. To increase the readability, the term “instance of” has been omitted in most places throughout this part of IEC 61850.

Services for the output model may have an impact on an internal process only, may produce an output signal to the process via a process interface, or may change a state value of a data attribute triggering a report. If the process interface is an IED in conformance with the IEC 61850 series, this service will produce an output signal to the process directly.

NOTE 2 The terms “input” and “output” are relative to the direction from the IED to the process (output) and from the process to the IED (input).



IEC 1414/11

The numbers in the circles in this figure are used in 6.4.2, 6.4.3 and related figures as references for the description.

Figure 13 – Output and input model (principle)

Several services are defined for the input model. The services communicating input information may carry information directly from the process interface or may have been computed inside an IED.

There are also several services that may be used to remotely manage the IED to some (restricted) degree, for example, to define a data set, to set a reference to a specific value, or to enable sending specific reports by a report control block. The information models (logical nodes and data classes) and the service models (for example, for reporting and logging) provide means to retrieve comprehensive information about the information model and the services that operate on the information models (self-description).

The following description of the output and input models are conceptual only. Details on the information and services involved in the models are defined in IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2.

6.4.2 Output model

6.4.2.1 Control model concept

The concept of the control model is depicted in Figure 14. The example is a circuit breaker logical node (XCBR) with the controllable data object XCBR.Pos (shown in Figure 15). A control service request is issued to the controllable data object. The service request contains service parameters like the control value, the originator of the request, the time when the originator sent the request and others. The data attribute XCBR.PosctlModel (shown in Figure 15) indicates the type of control service to use. Before the right control service request performs the change of the position of a real device, some conditions have to be met, for example, the output can be generated only if the local/remote switch is in the “remote” position and the interlocking node (CILO) has released this operation. The chain of conditions to be met may possibly include:

- the local/remote behavior of the circuit breaker (data object XCBR.Loc) and/or local/remote behaviour of the logical device (LD) (data object LLN0.Loc),
- the control authority condition at the station level (data object LLN0.LocSta),
- the control output signal is not being blocked, either by the process or:
 - by the mode of the circuit breaker (data object XCBR.Mod) or,
 - following an external control request (data object XCBR.BlkOpn and/or XCBR.BlkCls),

NOTE For controllable data in LNs not having specific data like BlkOpn and BlkCls, the blocked control output signal indication could be CmdBlk.

- check conditions of the device, and
- other attributes of the controllable data, for example, interlocking, pulse configuration, control model, sbo class, and sbo timeout as defined in the common data class DPC (controllable double point in IEC 61850-7-3).

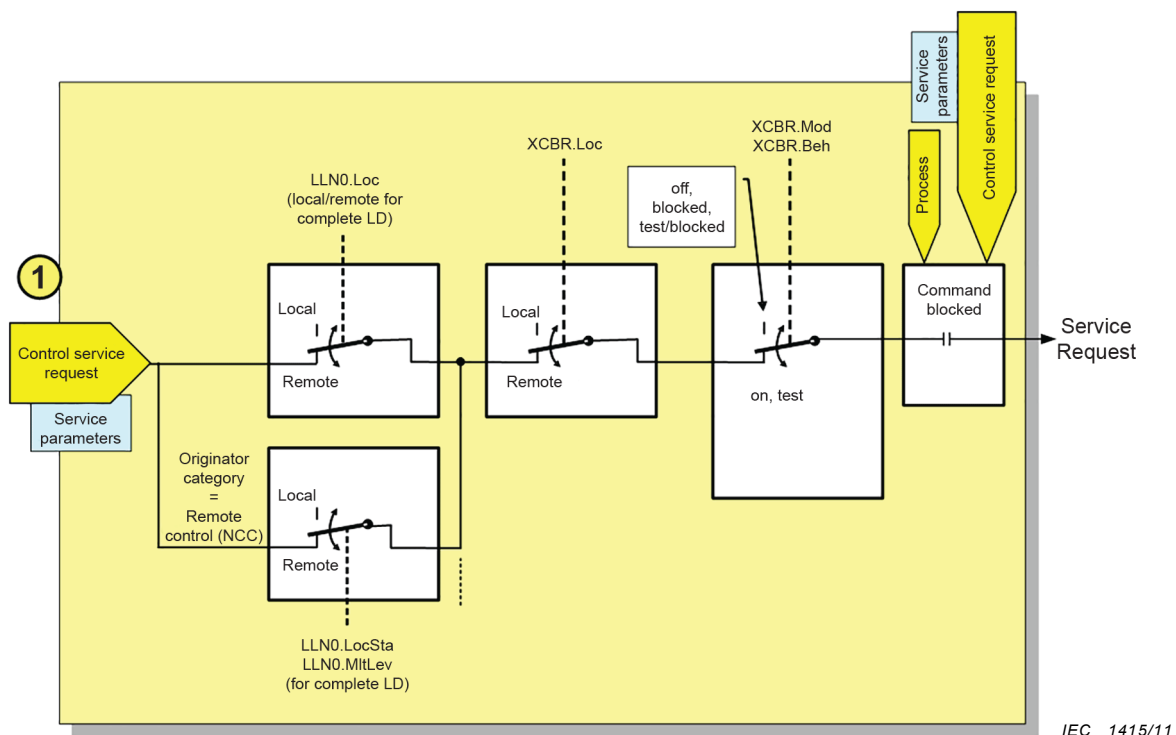
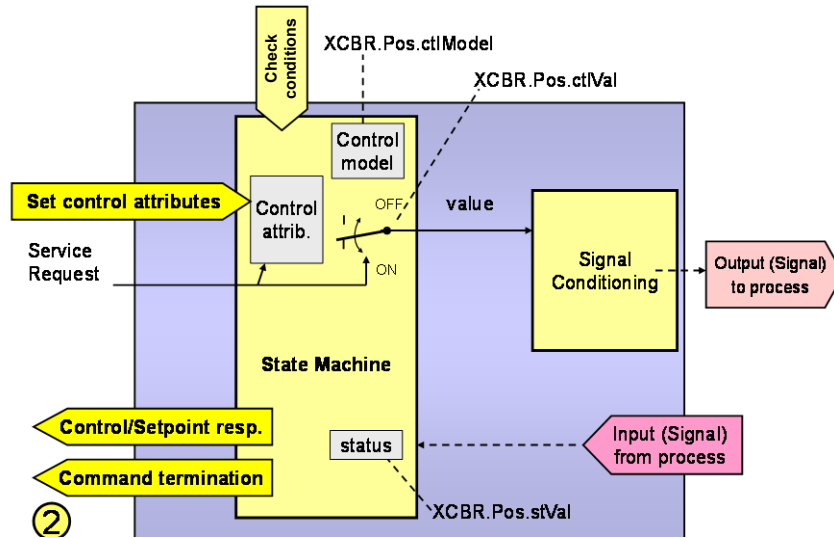


Figure 14 – Output model (step 1) (conceptual)

After all conditions have been met and all checks are positive, the output signal can be conditioned and control the real equipment (the circuit breaker – not shown).



IEC 1416/11

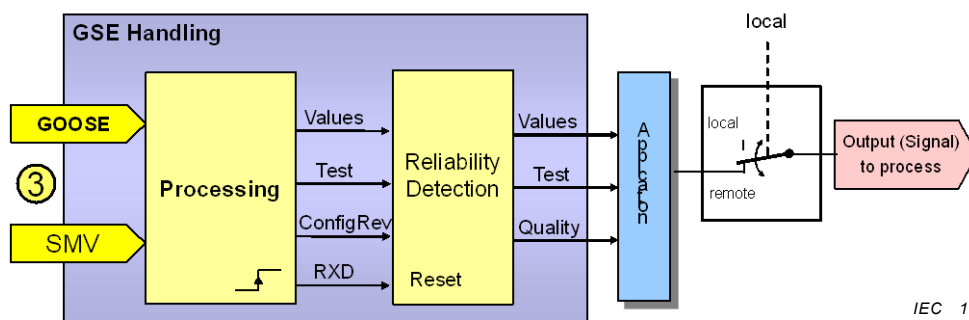
Figure 15 – Output model (step 2) (conceptual)

The state change of the real circuit breaker causes a change in the status information modelled with the data attribute XCBR.Pos.stVal. The status change issues a control service response. A command termination completes the control transaction.

6.4.2.2 GSE and SMV model concept

The generic substation event (GSE – GOOSE) and the sampled measured values (SMV), as shown in Figure 16, provides the peer-to-peer information exchange between the input data values of one IED to the output data of many other IEDs (multicast). The GOOSE and SMV messages received by an IED may be used to compute data for internal purposes also. An example for internal purposes are received switch positions to calculate the interlocking conditions locally or line current sample values to calculate the fundamental or RMS values.

NOTE 1 The GOOSE/SMV data values are defined in the input model described in 6.4.3.



IEC 1417/11

Figure 16 – GSE output model (conceptual)

The conditions to be met and the checks to run before the values are used as output signals such as interlocking are partly described within the IEC 61850 series and partly defined by the local application outside the scope of the IEC 61850 series.

The act of receiving GOOSE and SMV messages by an IED is called subscription. How to define subscriptions by means of configuration tools is given in the normative Annex H.

NOTE 2 Many GOOSE messages may be transmitted in certain cases, for example, fault detected by a protection relay. A SCSM usually filters these messages at the data link layer to prevent flooding the IEDs.

6.4.2.3 Attributes of data and control blocks

Many data attributes of the hierarchical information model can be set with a Set-service, for example, SetDataValues and SetDataSetValues. Setting the values of data attributes is usually constrained only by the application.

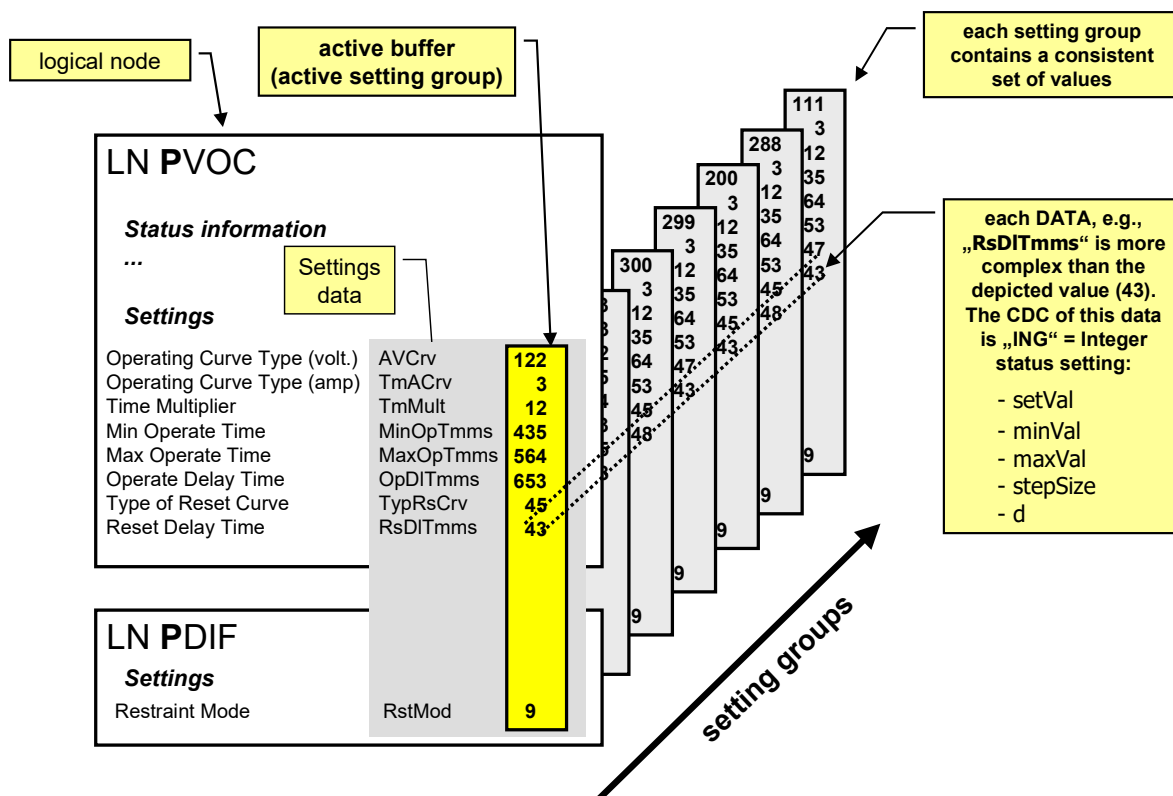
The various control blocks, for example, the setting group control block (SGCB), the buffered report control block (BRCB) and log control block (LCB), have control block attributes that can usually be set to a specific value. The services to set these attributes are defined with the control blocks in IEC 61850-7-2. Setting the values of the control block attributes is constrained by the state machine of the corresponding control block.

The control blocks behave according to the values of their attribute set. The values may also be configured using the SCL file or by other local means.

All control block attributes can be read by another IED.

6.4.2.4 Setting data and setting group control block

A special treatment of output data values is required for setting data contained in several logical nodes as defined in IEC 61850-7-4, for example, the settings for the voltage controlled overcurrent protection logical node PVOC (see Figure 17). The setting data (for example AVCrv, TmACrv, TmMult, etc.) have as many values as setting groups are defined. Each setting group has a consistent set of values.



IEC 1418/11

Figure 17 – Setting data (conceptual)

The values depicted are complex in the sense that each data has a type derived from a common data class. The RsDITmms is derived from the common data class ING. The ING has several data attributes as listed in Table 3.

Table 3 – Excerpt of integer status setting

ING class					
Attribute name	Attribute type	FC	TrgOp	Value/value range	PresCond
DataAttribute for configuration, description and extension					
<i>setting</i>					
setVal	INT32	SP	dchg	The value of the status setting.	M
setVal	INT32	SG, SE		The value of the status setting.	M
<i>configuration, description and extension</i>					
minVal	INT32	CF	dchg	Minimum setting for 'setVal'.	O
maxVal	INT32	CF	dchg	Maximum setting for 'setVal'.	O
stepSize	INT32U	CF	dchg	(range=[1...(maxVal-minVal)]) Step between the individual values of 'setVal'.	O
units	Unit	CF	dchg	Unit for 'setVal', 'minVal', 'maxVal', 'stepSize'.	O
d	VISIBLE STRING255	DC		Textual description of the data. In case it is used within the CDC LPL, the description refers to the logical node.	O
...

The values of a specific setting group contained in the setting data can be set only if that group is in the “EDIT” state (indicated by the FC=SE; edit setting data). After all values of that group are set, the values of that group can be confirmed as containing a consistent set of values. This newly confirmed set of values can then be selected for use by the application (setting group in active state: FC=SG; active setting data).

The setVal of FC=SP means “simple” setting data (set point); applied when the setting group control model is not supported. This value can be set as a regular data attribute.

6.4.3 Input model

6.4.3.1 Input analogue signal acquisition

The concept of the input analogue signal acquisition is depicted in Figure 18. Normally, the signal is conditioned by a signal conditioner. In the IEC 61850 model, an analogue input does not exist as data before it is converted from analogue to digital. The sample rate (data attribute smpRate of a configurable data) determines how often the value shall be sampled. Alternatively, the raw digital values may be directly obtained from sample values transmitted over communication links (see also Figure 27). The method used for processing the signal (True RMS, Peak amplitude, ...) may be set through the data ClcMth. In absence of the data ClcMth, the calculation method shall be considered as UNSPECIFIED, meaning UNKNOWN.

The conditions to be met before the value can be communicated (modelled as the data attribute instMag of the data, for example, a voltage of a specific phase – see Figure 18) may comprise the values of the following attributes:

- substitute/unsubstitute “switch” of the data (modelled as the data attribute subMag of the data, for example, a voltage of a specific phase),
- operator blocked or unblocked “switch”.

The result of these first steps is the “intermediate value” (still an analogue value) accompanied by the corresponding quality information.

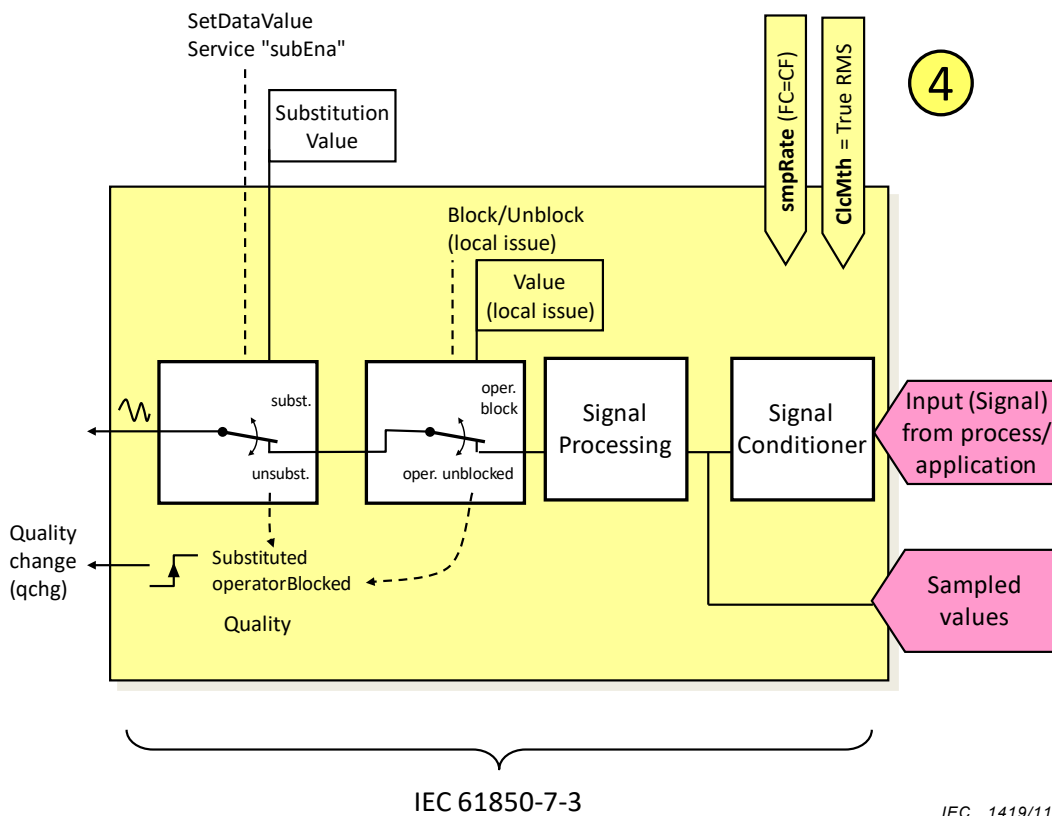


Figure 18 – Input model for analogue values (step 1) (conceptual)

6.4.3.2 Data attribute value processing, monitoring and event detection

The “intermediate value” is used for various purposes. As shown in Figure 20, the first use is to provide this value as the instantaneous data attribute value (magnitude) of the data. The data attribute has the name *instMag*; with the functional constraint FC=MX (indicating a measurand value). There is no trigger option associated with the instantaneous value.

The second application is the calculation of the deadbanded value, the *mag* value. The deadbanded value shall be based on a deadband calculation from *instMag* as illustrated in Figure 19. The value of *mag* shall be updated to the current value of *instMag* when the value has changed according to the value of the configuration parameters *db* and *dbRef* of this data.

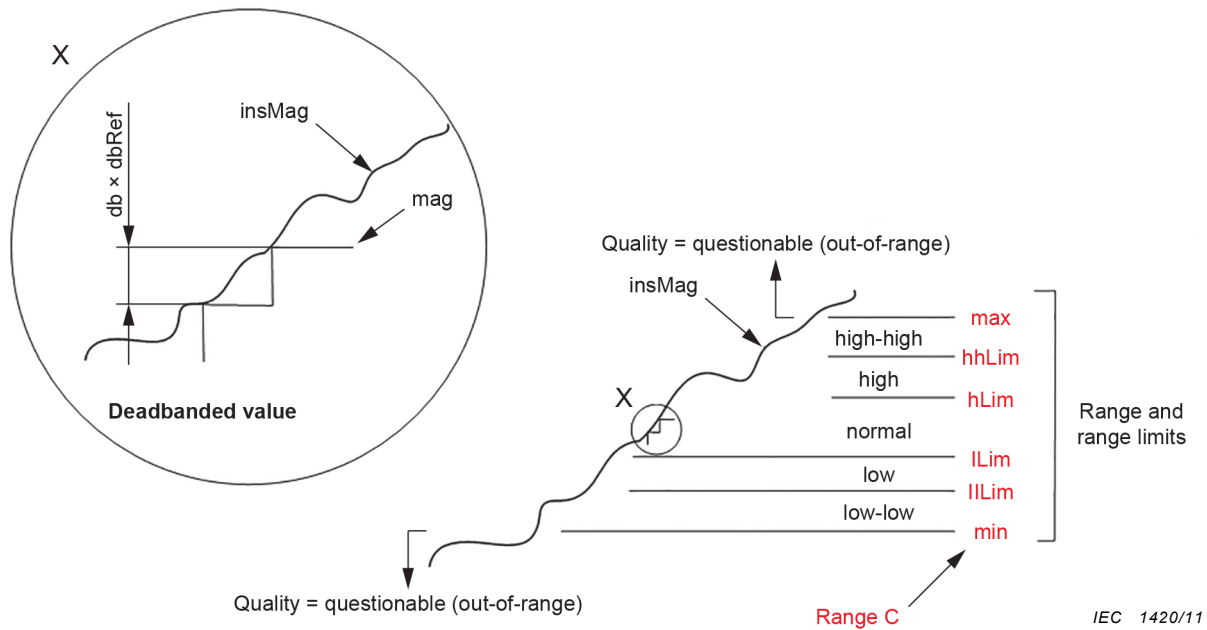


Figure 19 – Range and deadbanded value (conceptual)

The value of the deadband configuration db shall represent the percentage of the deadband reference $dbRef$ in units of 0,001 % (fixed deadband). A value of $dbRef$ of 0 means that the value db shall represent the percentage of the last transmitted values in units of 0,001 %. This allows for a deadband calculation that is related to the last refreshed value (variable deadband).

NOTE The db value has nothing to do with the accuracy of the data defined both by the accuracy of the analogue transducer and by the accuracy of the A/D conversion.

An internal event is created any time the mag value changes. The deadbanded value mag and the event (data change – according to the trigger option $TrgOp=dchg$) are made available for further actions, for example, reporting or logging.

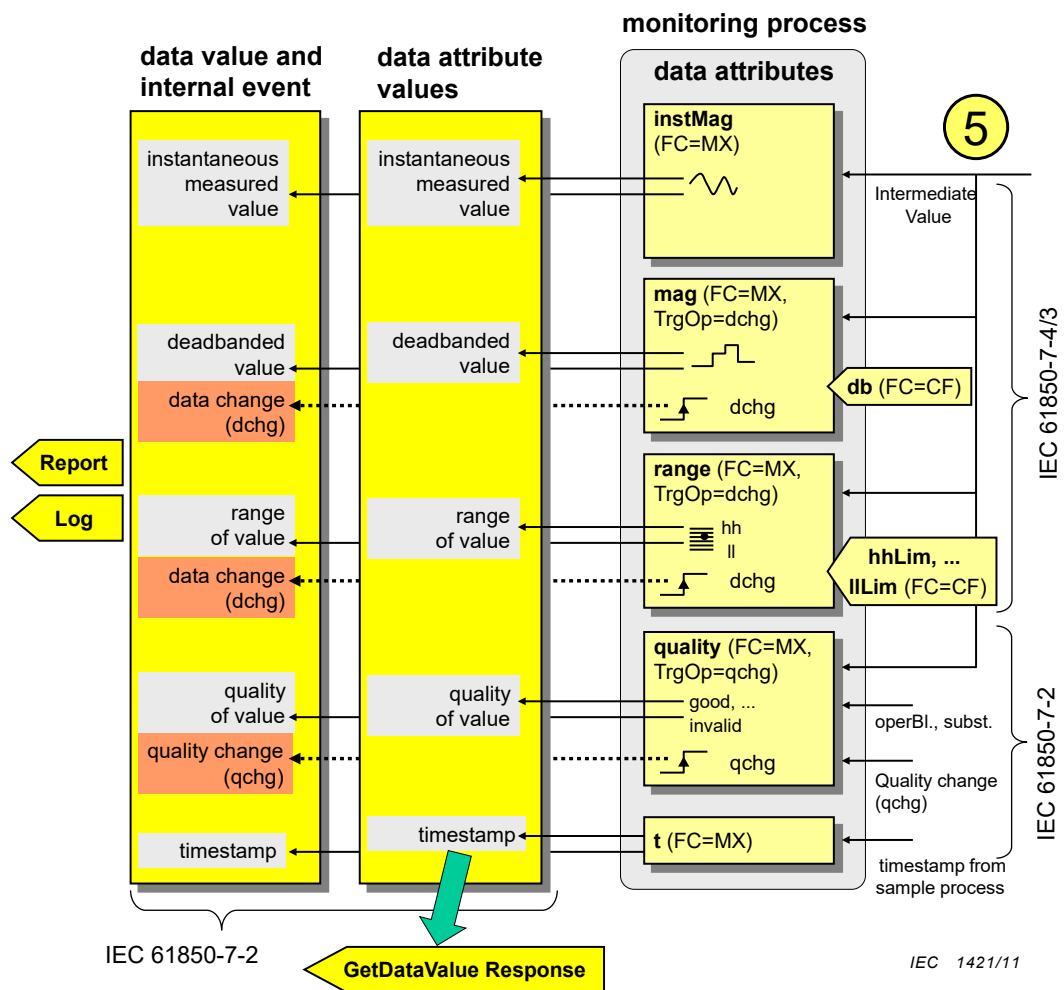


Figure 20 – Input model for analogue values (step 2) (conceptual)

A third application is to monitor the “intermediate value” to determine the current range of the value.

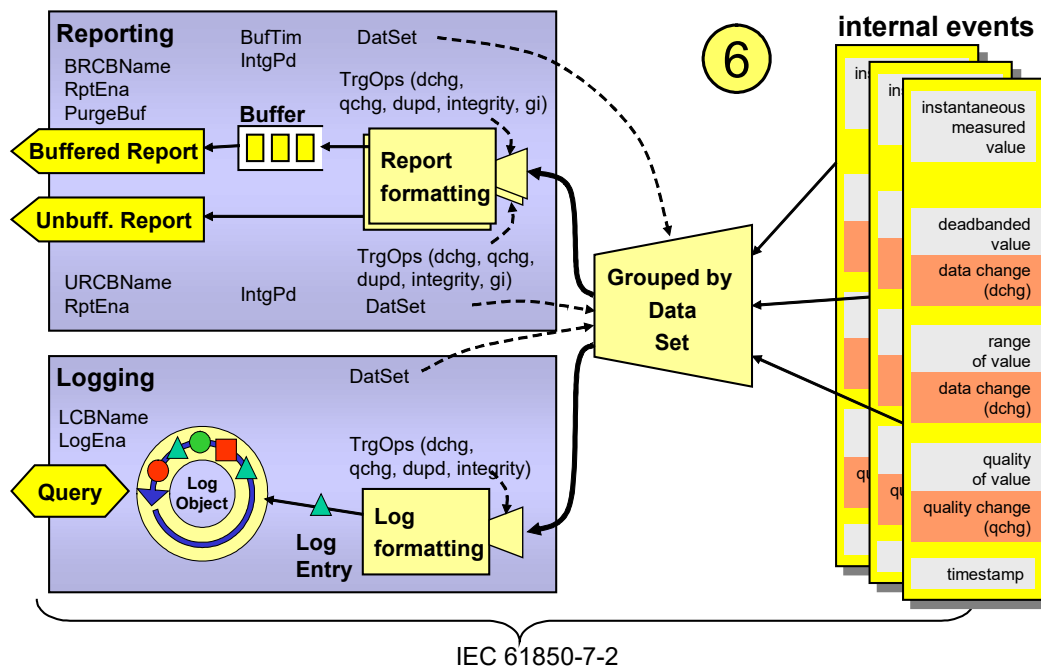
An internal event is created any time the *instMag* value crosses a range limit (see Figure 20). The range value and the event (data change – according to the trigger option *TrgOp=dchg*) are made available for further actions, for example, reporting or logging. The attribute *rangeC* is used to configure the parameters associated with the different limits.

In addition to the various values, the two attributes *quality* and *t* (time stamp) are available at any time. The time stamp is determined at the time that the value change of the data attributes *mag* and *range* has been detected and/or at the time a change of the *quality* has been detected. A change in the *quality* can be used to issue an internal event as well.

6.4.3.3 Data reporting and logging

6.4.3.3.1 General

The internal events (process values, corresponding trigger values that caused the event, time stamps and quality information) are used as a trigger foundation for reporting and logging (see Figure 21). This information is grouped using a data set. The data set is the content basis for reporting and logging. It is also the content basis for GOOSE and SMV messages (see 6.4.3.4 and 6.4.3.5). The data set contain references to the data and data attribute values.



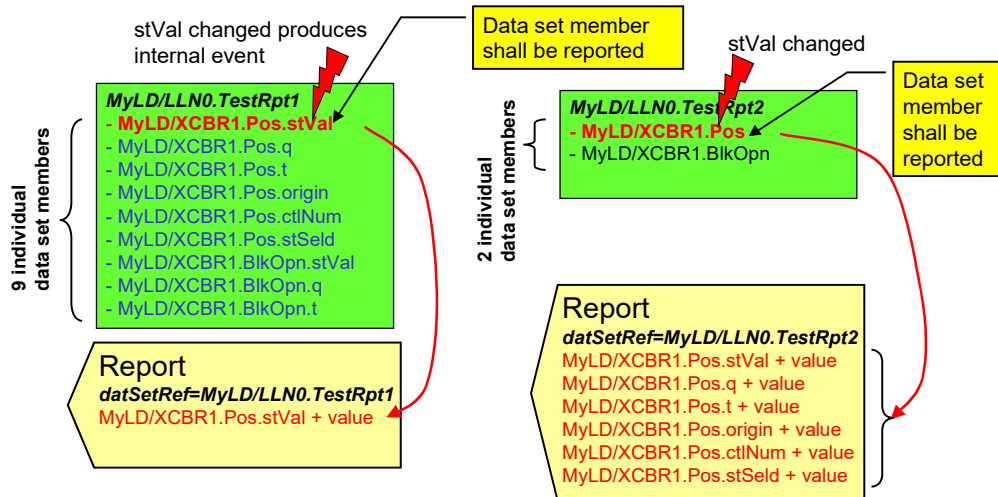
IEC 1422/11

Figure 21 – Reporting and logging model (conceptual)

Which data and data attribute values are to be reported and logged is specified in the data sets. The following example explains the concept.

6.4.3.3.2 Data reporting

The data attribute stVal of the data MyLD/XCBR1.Pos (Position) in Figure 22 is referenced in two different data sets. The figure displays two different instances of data sets that reference the data attributes of the position. In the case on the left, the data set references 9 individual data set members (all of functional constraint ST): Pos.stVal is one of the nine members. In case of the change triggered by the member stVal, the value for exactly that member shall be included in the report. The data set in the example on the right-hand side has just two members. The data Pos (which has six data attributes: stVal, q, t, etc.) is one of the two members. A change triggered in the member Pos (for example, by the change in the DataAttribute stVal) shall cause the inclusion of the values of all data attributes of the data set member Pos (i.e., the complete member comprising all six data attributes stVal, q, t, etc.).



NOTE All data attributes are functionally constrained by FC=ST.

IEC 1423/11

Figure 22 – Data set members and reporting

The data set specifies which data is to be monitored and reported. The next task is to define when and how to report or log the information. The reporting model provides two kinds of report control blocks:

- 1) the unbuffered, and
- 2) the buffered control blocks.

The log model has the log and log control block.

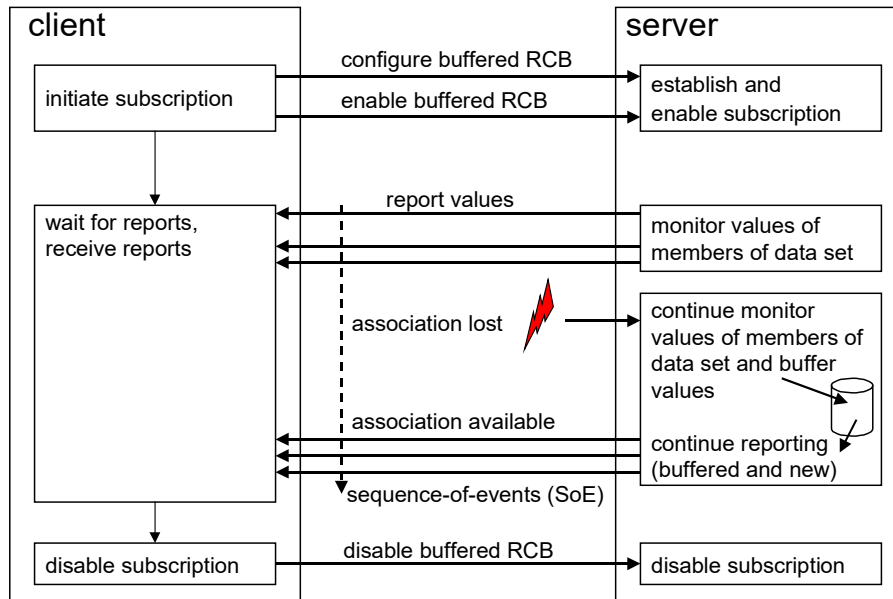
The principle characteristics of the data access methods provided by IEC 61850-7-2 are shown in Table 4.

Table 4 – Comparison of the data access methods

Retrieval method	Time-critical information exchange	Can lose changes (of sequence)	Multiple clients to receive information	Last change of data stored by	Typical client (but not exclusive)
Polling (GetDataValues)	NO	YES	YES	–	Browser
Unbuffered Reporting	YES	YES	NO	–	Real-time GUI
Buffered Reporting	YES	NO	NO	Server	Data concentrator
Log (used for SOE logging)	NO	NO	YES	Client	Engineering stations

Each of the four retrieval methods has a specific characteristic. There is no single method that meets all application requirements. During system design, the designer has to analyse the requirements and to check them against the (implemented) methods provided by a device compliant with the IEC 61850 series.

The basic buffered reporting mechanism is shown in Figure 23. The buffered and unbuffered reporting starts with the reservation of the report control block by the client, followed by configuration of the report control block, including the verification that the report control block parameters are meeting the needs of the client. The reporting to the client starts with setting the enable buffer attribute to TRUE; setting to FALSE stops the reporting to the client.



IEC 1424/11

Figure 23 – Buffered report control block (conceptual)

The specific characteristic of the buffered report control block is that it continues buffering the event data as they occur according to the enabled trigger options in case of, for example, a communication loss. The reporting process continues as soon as the communication is available again. The buffering is independent from the reporting being enabled or not. The buffered report control block guarantees the sequence-of-events (SoE) up to some practical limits (for example, buffer size and maximum interruption time).

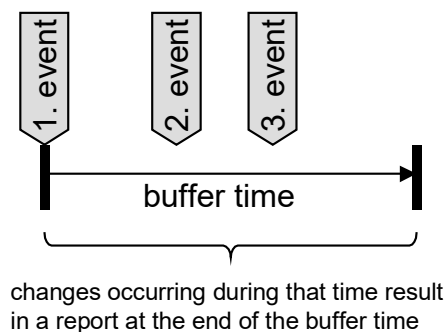
The unbuffered report control block does not support SoE in case of loss of communication.

The buffered report control block has several attributes that control the reporting process, for example:

- RptID handle provided by the client to identify the buffered report control block,
- RptEna to remotely enable/disable the reporting process,
- DataSet references the data set whose values are to be reported,
- ConfRev contains the configuration revision to indicate addition/deletion of a member of the data set or the reordering of the members,
- OptFlds indicates the optional fields which are to be included in the report; among the optional fields are:
 - sequence-number to get the correct order of events,
 - report-time-stamp to inform the client when the report has been issued,
 - reason-for-inclusion to indicate the trigger that has caused the value to be reported,
 - data-set-name to indicate from which data sets the values have been generated,
 - data-reference to include the object references for the values,
- BufTm contains the time to wait after the first event within a data set has occurred before issuing the report (see Figure 24),
- SeqNum is the current sequence number of the reports,
- TrgOps (trigger options) contains the reasons which cause the control block to report a value into the report. The reasons for reporting may be: data change dchg, data update dupd, or quality change qchg of the data attribute in a logical node,

- IntgPd (integrity period): reporting all values initiated by the server based on this period,
- GI (general interrogation): reporting all values initiated by the client, and
- PurgeBuf set to TRUE indicates to delete all events not yet sent.
- ResvTms indicates if the the buffered report control block is reserved or not by a set of specific clients based upon configuration. The value represents the number of seconds that the reservation will be maintained after association loss.

If it is likely that – after a first event – several other events occur in the direct neighbourhood of the first event (see Figure 24), then the server can reduce the number of reports applying the buffer time attribute. Changes occurring during that time result in a report at the end of the buffer time reporting all changes (according to the reasons and according to the definition of the corresponding data set for a specific report control block).



IEC 1425/11

Figure 24 – Buffer time

A report allows sending just the values (according to the reason and according to the definition of the corresponding data set for a specific report control block) without any object reference of the data and data attributes. Then the object references may be retrieved out of the data set definition (see below). The report may also transmit the object references of the data and data attributes together with the data.

If firstly, no object references are sent with the values and secondly, values for a subset of members of a data set only are to be reported, then a provision is provided to determine to which members the reported values belong. The SCSM defined in IEC 61850-8-1 defines an inclusion-bitstring to indicate the member of the data set. The order of the members of the data set is as they are defined in the data set. Figure 25 shows an example.

The data set has two members in the order shown. The report with the inclusion-bitstring has two bits whose values indicate from which members the values have been derived. The first bit is TRUE thus indicating that the values in curly brackets are the values from the member MyLd/XCBR1.Pos. For the second member, no values are reported (second bit is FALSE). The report with the inclusion-bitstring optimises the length of the report message.

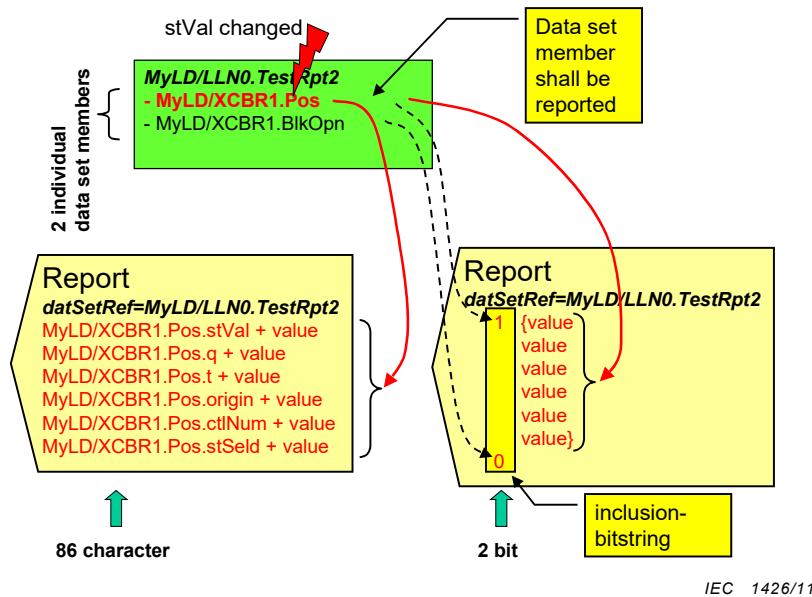


Figure 25 – Data set members and inclusion-bitstring

6.4.3.3.3 Data logging

The logging model provides a log to store values (the log entries). The log control block controls which data values and when these data values are to be stored into the log. The log is organised as a circular buffer as shown in Figure 21. The number of entries that can be stored depends on the size of log entries and on the buffer size.

NOTE Several factors have an impact on the design of a log. The system needs to be designed carefully to implement or configure the log and log control blocks in a way that meets the application requirements. Recommendations with regard to the system design are outside the scope of this standard.

Figure 26 shows an example of a log and three log control blocks. The first step is to configure and enable log control blocks. After enabling the association with that server may be closed. The log entries are stored into the log as they arrive for inclusion into the log. The logs are stored in time sequence order. This allows retrieval of a sequence-of-events (SoE) list.

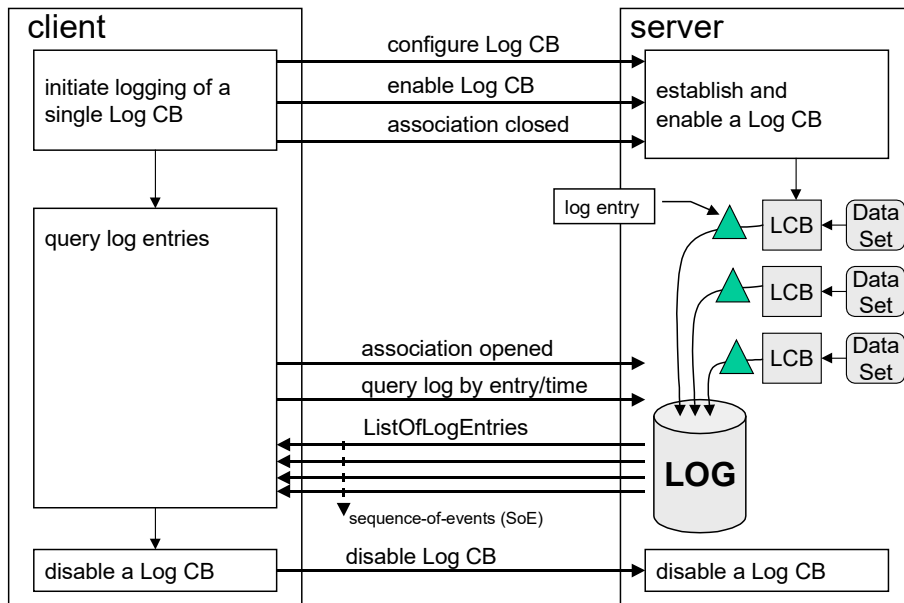


Figure 26 – Log control block (conceptual)

The log (not the log control block) is enabled at any time. The different log control blocks allow storage of information from different data sets into the log. Each log control block is independent of the other control blocks.

The log control block has several attributes that control the logging process, for example:

- LogEna to remotely enable/disable the logging process,
- DatSet references the data set whose values are to be logged,
- TrgOps contains the reasons which cause the control block to store an entry into the log. The reasons for storing a log entry into the log may be: data change dchg, data update dupd, or quality change qchg of the data attribute in a logical node,
- integrity period IntgPd: logging all values initiated by the server based on a given period,
- LogRef indicating in which log the entries are to be stored.

6.4.3.4 Peer-to-peer data value publishing

The peer-to-peer communication provides services for the exchange of generic substation events (GOOSE based on multicast) and for the exchange of sampled values (based on multicast or unicast). The GOOSE message receipt is already explained in the output model in 6.4.2.2.

Figure 27 shows the GOOSE and sampled value models.

The GOOSE model uses data values to be published grouped into data sets. All data and data attributes can be used to create a data set (for example, analogue, binary or integer values).

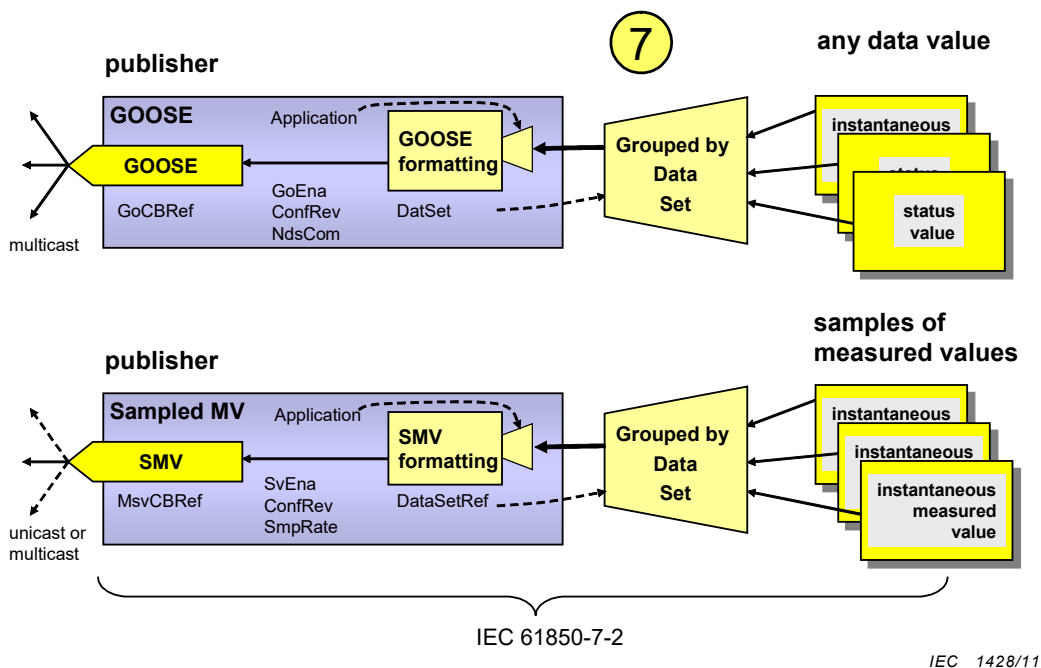


Figure 27 – Peer-to-peer data value publishing model (conceptual)

The GOOSE model has several attributes that control the publishing process, for example:

- GoEna to remotely enable/disable the publishing,
- GoID send in the message to be used as a handle for the receiving application,
- DatSet references the data set whose values are to be published,
- ConfRev contains the configuration revision to indicate deletion of a member of the data set or the reordering of the members, or changing the DatSet reference,

- NdsCom indicates in the message that some commissioning is required.

Which event triggers the publishing of values as well as how often and how fast the values are to be published is a matter of the requirements of the function being implemented.

6.4.3.5 Sampled value publishing

The sampled value publishing model has several attributes that control the publishing process, for example:

- SvEna to remotely enable/disable the publishing,
- MsvID sent in the message to be used as a handle for the receiving application,
- DataSet references the data set whose values are to be published,
- ConfRev contains the configuration revision to indicate deletion of a member of the data set or the reordering of the members, or changing the DataSet reference,
- SmpRate specifies the samples per unit.

6.4.4 Model for statistical and historical statistical data

In many application domains, it is required to provide additional information of a basic analogue value:

- statistical information (for example, minimum value calculated for a specified time period, for example, minimum value of last 1 h),
- historical statistical information (for example, log of minimum values of the sequence of values calculated above, for example, last 24 hourly values).

This additional information may be derived from the basic analogue values. It may be the only information provided – depending on the application requirements.

The models for the statistical and historical statistical data are explained conceptually in Figure 28. On the left hand side are the basic data representing the current values (UNSPECIFIED), i.e. some instantaneous analogue (or integer) values that are contained in the logical node instance XXYZ.

Instantaneous values

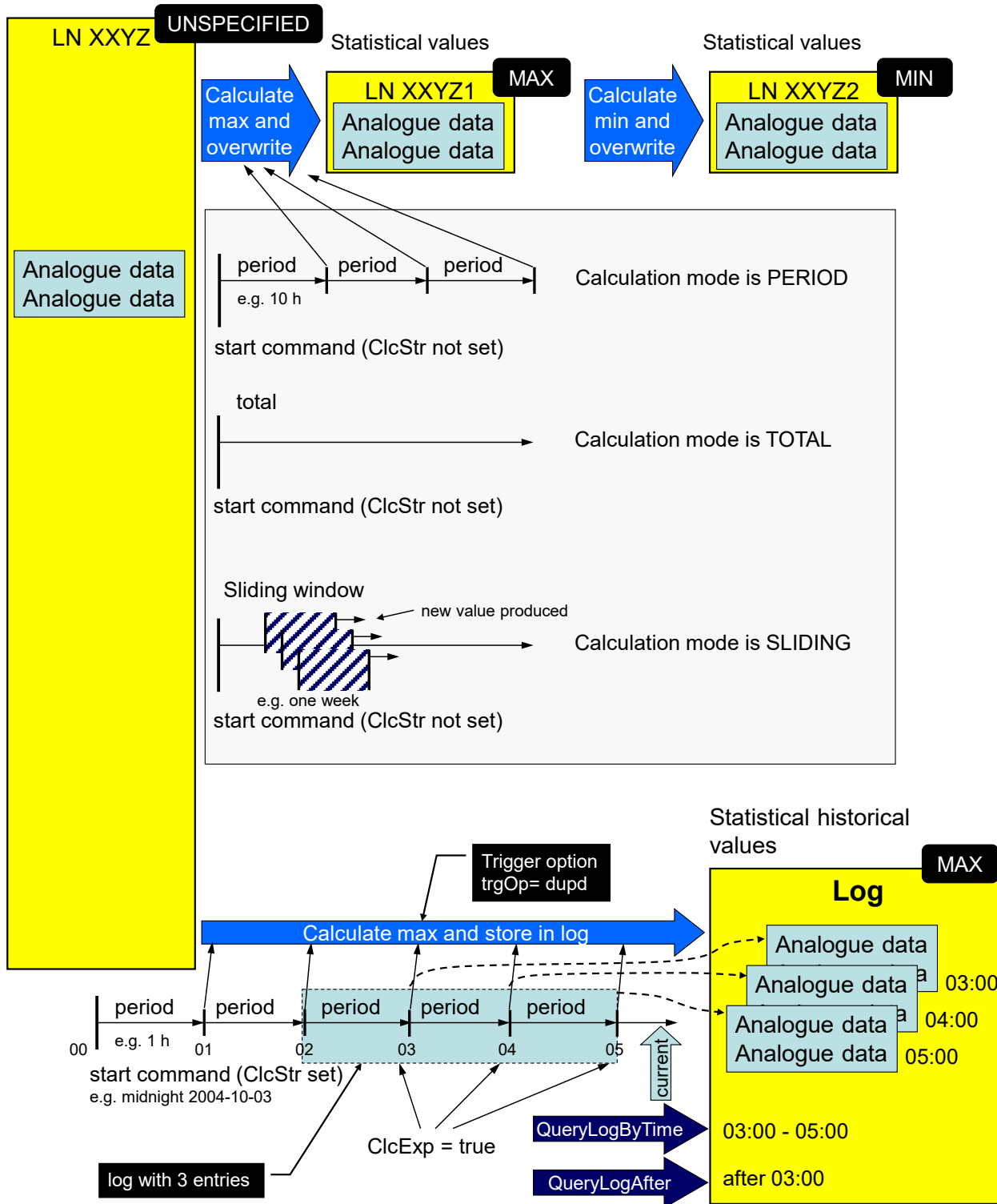


Figure 28 – Conceptual model of statistical and historical statistical data (1)

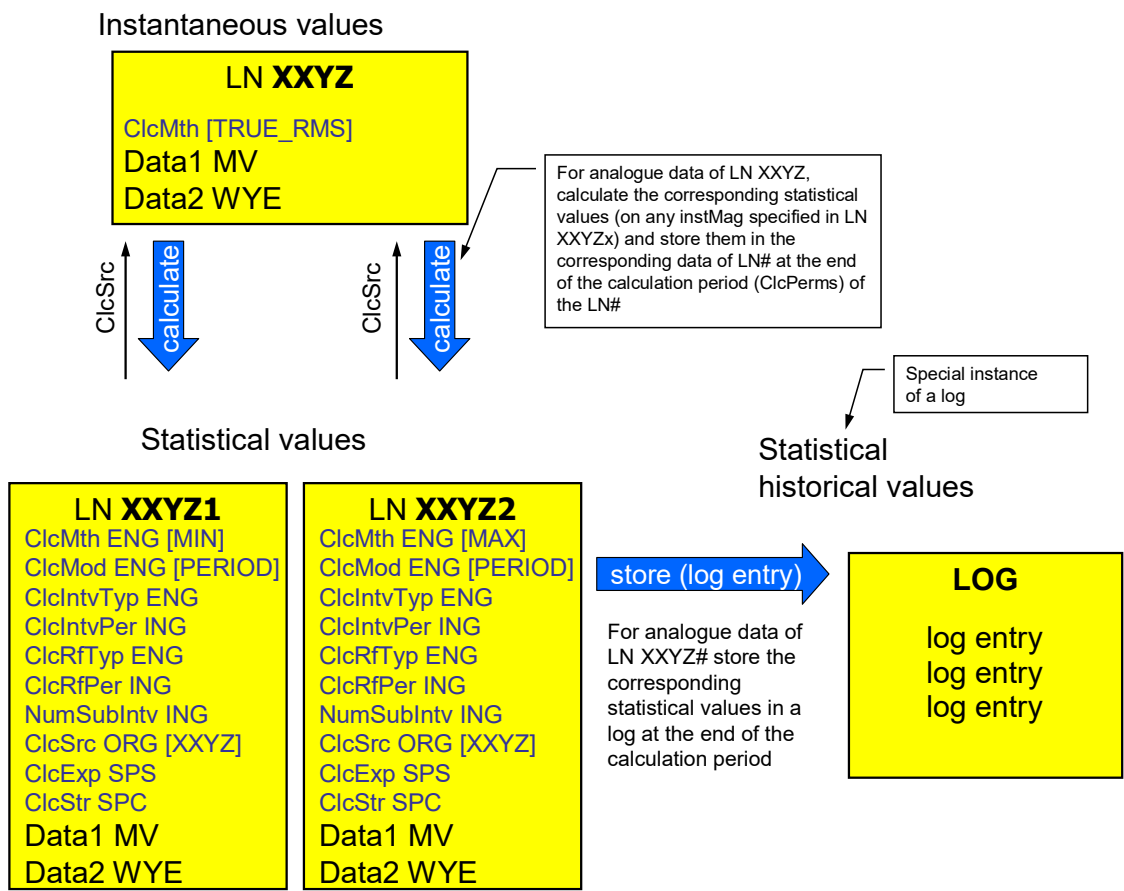
The upper half depicts the method defined for statistical values. The first example is the instance XXYZ1 of the logical node class XXYZ. The analogue values represent the calculated maximum values derived from the instance XXYZ. The logical node XXYZ1 has special setting data that indicate that the values are maximum values and that the calculation method is “periodic”. The period starts after a start command. At the end of the period, the calculated maximum values of the instance XXYZ1 are overwritten by the new values.

The maximum values can be used to calculate the minimum maximum values in – of course – a much longer period than for the maximum calculation in XXYZ1. The instance XXYZ2 may represent the minimum value of the max value of the last 10 days.

Setting parameters other than PERIOD may be used to specify calculation modes. A calculation mode set to TOTAL means that the calculation of maximum values will start as soon as it has been enabled. A calculation mode set to SLIDING means that the calculated maximum values are calculated over a sliding window whose width can be set by means of a special interval type setting (e.g. hour, day, week).

The lower part of the figure shows the conceptual model of the historical statistical data. In this model, the values are calculated periodically (in this case the maximum values with calculation mode set to PERIOD) and are stored in sequence in a log. The calculation in the example starts at midnight of 2004-10-03. The interval is 1 h. After the first hour, the first log entry is written. After the second hour, the second entry contains the value of the second hour. After 5 h, the log contains the values of the last three hours (intervals 02-03, 03-04, 04-05). The expiration of the statistical calculation is indicated by the “event” ClcExp set to true and at this moment the statistical values that have the trigger option dupd are logged even if the values have not changed. The same concept can be applied to the reporting of statistical data.

The statistical data model is based on the calculation of analogue values contained in other logical nodes. Figure 29 comprises three technological logical nodes of the same type (for example MMXU). The top logical node (LN XXYZ) represents the RMS measured values. The second and third logical nodes are the statistical logical nodes, i.e., the logical nodes that represent the calculated values (LN XXYZ1 represents the MIN values, the LN XXYZ2 the MAX values).



IEC 1430/11

Figure 29 – Conceptual model of statistical and historical statistical data (2)

The two logical nodes on the bottom left in Figure 29 (XYZ1 and XYZ2) represent minimum (MIN) and maximum (MAX) values of the analogue data represented in the top logical node (XYZ). Minimum and maximum values are defined by the setting ClcMth. The two logical nodes make use of the setting data ClcSrc (calculation source). The common data class of ClcSrc is ORG, "object reference setting group" and is used to reference the source logical node for the calculation. For both logical nodes, ClcSrc has the value XYZ. Each logical node with analogue data can be used as a source.

The lower part of the figure shows the conceptual model of the historical statistical data. In this model, the values are calculated periodically (in this case the maximum values with calculation mode set to PERIOD) and are stored in sequence in a log. The calculation in the example starts at midnight of 2004-10-03. The interval is 1 h. After the first hour, the first log entry is written. After the second hour, the second entry contains the value of the second hour. After 5 h, the log contains the values of the last three hours (intervals 02-03, 03-04, 04-05). The expiration of the statistical calculation is indicated by the "event" ClcExp set to true and at this moment the statistical values that have the trigger option dupd are logged even if the values have not changed. The same concept can be applied to the reporting of statistical data.

6.4.5 Model for system functions

6.4.5.1 General

System specific information is modelled using logical nodes from the group L. This subclause gives a short description of the functions intended to be modelled by those logical nodes.

6.4.5.2 Time synchronization

The time synchronization model shall provide accurate time to all IEDs in a power utility system for data time stamping with various ranges of accuracy, e.g. millisecond range for reporting, logging and control and microsecond range for sample values.

The different components of the time synchronization model are:

- an external time source with a known level of accuracy (for example, GPS),
- a time server providing the internal time source for all IEDs in the substation,
- a time synchronization protocol providing synchronization between IEDs (for example SNTP for IEC 61850-8-1),
- the time stamp semantics used for the information exchange defined in the ACSI,
- the presentation of the time stamps according to the SCSM,
- the server and clients that need substation-wide synchronized time.

The chosen reference for time stamping is UTC. Using UTC guarantees that time evolution is continuous and is independent of abrupt changes due to daylight savings time (DST). This is very convenient for applications which have to sort events by date and time.

6.4.5.3 Time management

At the user level, it is generally necessary to display local time according to the time zone and DST. The standard considers this translation a local application issue. It is also needed to manage the two moments when DST changes. Time management functions are modelled by means of the logical node LTIM. LTIM provides data objects (settings) in order to read or change the local time parameters depending on the IED capabilities. If an IED is required to display information using local time but does not internally manage any time zone table, this can be handled by a central IEC 61850 client in the system which knows the time zone table parameters.

6.4.5.4 Time supervision

Time supervision is the function responsible for communicating the status of the time synchronization function of an IED. The logical node LTMS has been introduced for that purpose. It allows to know important conditions with regard to time synchronization as the status of the communication channel with the time source or the time accuracy class according to IEC 61850-7-2.

6.4.5.5 GOOSE and SMV supervision

The LGOS and LSVS logical nodes may be used to monitor subscription states to GOOSE or SMV signals. They contain mandatory information like status of the subscription (active, not active) and other optional information e.g. the source GOOSE or SMV control block identification. The normative Annex G describes the configuration flow with regards to these supervision logical nodes.

6.4.5.6 Service tracking

Service tracking is represented by the logical node LTRK and is defined as the function in charge of recording, after a service execution, the parameters values used by any service defined in IEC 61850-7-2. It is thus possible to read, report or log these values for system behaviour analysis.

Figure 30 shows an example of a logical node LTRK used to track the execution of control services. In the example, an operate service with the value close is successfully executed on the object XCBR.Pos, i.e. the position of a breaker³. The service uses the parameters defined in IEC 61850-7-2 and is tracked by the logical node LTRK which contains the data class DpcTrk. The data class DpcTrk is used to record the last applied parameter values on any controllable object within the IED whose CDC is of type DPC (controllable double point).

³ The example shown in Figure 30 is conceptual and XCBR.Pos is used for convenience as the control object reference, which is incomplete. The control object reference of an instance shall contain the complete reference to an objectname including the logical device name as explained later in this document.

The logical node LTRK contains data classes needed for any service defined in IEC 61850-7-2. The data classes are based on common data classes (CDC) also defined in IEC 61850-7-2. In the example, DpcTrk contains attributes inherited from the CDC CTS, common service tracking common data class. These attributes are common to all service tracking data classes. DpcTrk also contains attributes which are specific to the control service tracking data classes. The attributes have the functional constraint FC=SR.

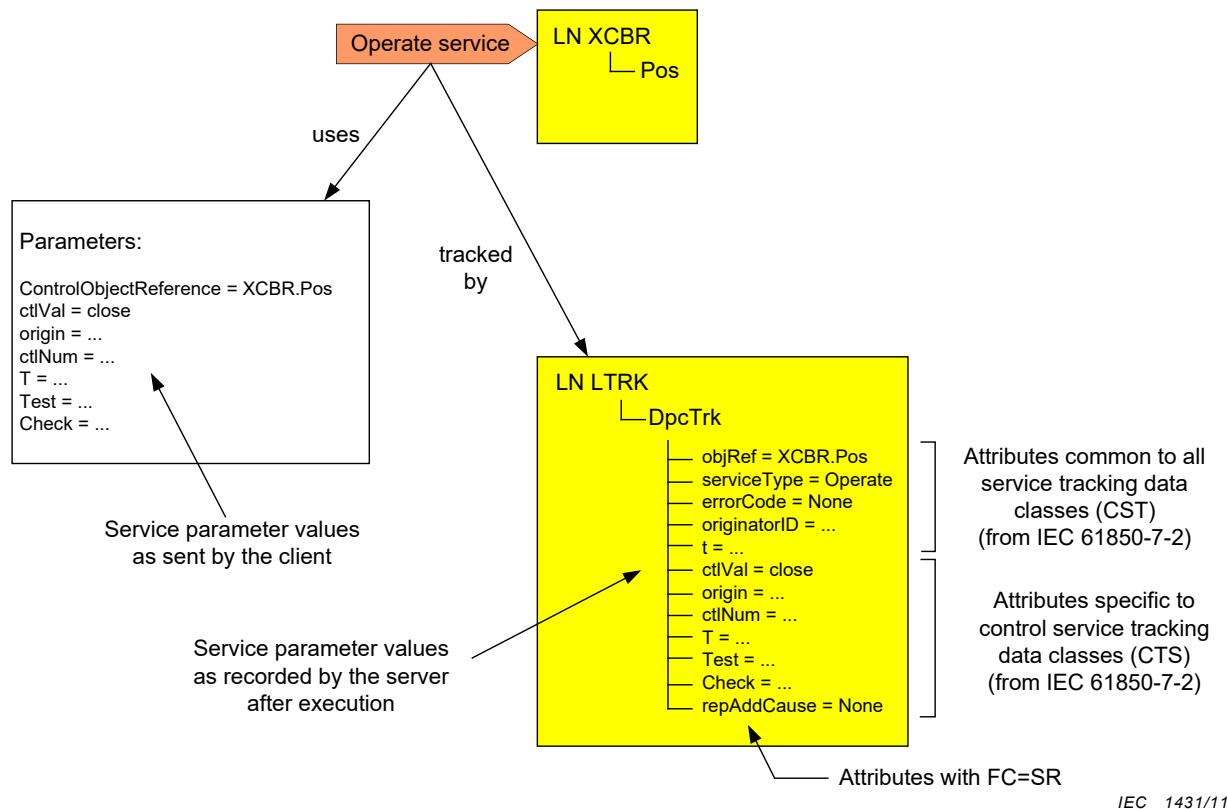


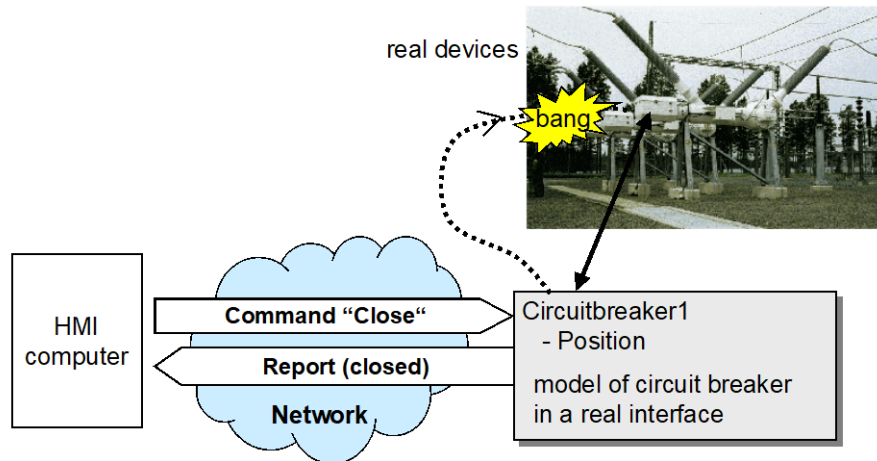
Figure 30 – Concept of the service tracking model – Example: control service tracking

7 Application view

7.1 General

A sample operation, illustrated in Figure 31 is to switch a circuit breaker. An operator at a remote HMI wants to remotely switch the circuit breaker. The HMI computer and the circuit breaker have to operate together (interoperate). First, the computer needs to know what information it has to transmit to the IED representing the circuit breaker (normally called the “process interface”). Secondly, it has also to know the name of this IED (for example “Circuitbreaker1”) and how to address the IED. Both the HMI computer on the left side and the IED “circuitbreaker1” on the right side are connected to a common communication network. The HMI sends a control command to the “Circuitbreaker1” to switch the position of the breaker (close the breaker). After switching is completed, the interface IED may (if configured) send a report to the HMI computer indicating that the switch position has changed.

Different users may name the circuit breaker differently: one may use “Circuitbreaker1”, another may chose “CBK-2”. IEC 61850-7-4 based on the approach described in IEC 61850-5 standardises many abbreviated names for substation functions and related equipment. The standardised name for a circuit breaker is XCBR. This name may be accompanied by a suffix and a prefix: “Q1XCBR1” (for naming conventions, see 12.4, Clause A.1, Clause A.2 and IEC 61850-7-2).



IEC 1432/11

Figure 31 – Real world devices

Applications in the computer on the left-hand side of Figure 31 may also query information about the:

- real physical circuit breaker (nameplate, health, ratings, etc.),
- real device (IED) that hosts the process interface (nameplate, health, operation mode, etc.),
- behaviour of the reporting services that determines the transmission of status reports.

In addition, the operator (or the computer if it is in some type of automatic mode) may change the active setting group of a protection function to a different setting group, may configure the reporting behaviour remotely, or may request the substitution of a fixed value for a process value. Alternatively, the operator may want to receive a sequence of events.

All these and many other functions supported by the controller have three major aspects in common that are standardised in the IEC 61850 series:

- what functions and what information is network-visible and how are they named and described (see IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2),
- how functions can be accessed and how (more generally speaking) information can be exchanged (see IEC 61850-7-2), and
- how devices can be connected to communication networks (IEC 61850-8-x and IEC 61850-9-x series).

IEC 61850-7-4 comprises a list of more than 2 000 named and well-defined information elements which enable creation of the information model of a real substation device (for example, complete power transformer, circuit switch, or measuring unit). This static information model inherits the type information from IEC 61850-7-3 and the required (dynamic) communication services from IEC 61850-7-2. Functions in the context of the IEC 61850-7-x series are those that are needed to exchange all information of the information model in the manner that is required for the substation domain or other domains within the scope of IEC technical committee 57. Functions (for example, substation functions like bus bar protection, or point of wave switching) make use of the data and functions provided by IEC 61850-7-x series.

From the point of view of the IEC 61850-7-x series, the internal processes involved to accomplish functions represented by logical nodes within an IED are outside the scope of IEC 61850-7-x series. Only the interactions between logical nodes located in different IEDs have to use the defined services and data. Examples of interaction of logical nodes for complex functions such as synchronised switching including the basic sequence of exchanged messages can be found in IEC 61850-5.

The dynamic behaviour of a real device is established through the configurable attributes of the implemented information model, and by changing its (changeable) values. The effects resulting from any change of value in the information model is defined in the standard. As a result of a control of the “Circuitbreaker1”, the real circuit switch opens or closes. The controller may immediately send a report (value, quality, and timestamp) to the initiator and may additionally write this event in the log of the device for later retrieval. Various dynamic behaviours of the controller may be pre-configured by an engineering tool. The behaviour may be changed by configuring the controller using specific services for remote configuration (set), for example, reporting values can be remotely enabled or disabled.

7.2 First modelling step – Logical nodes and data

IEC 61850-7-4 defines a list of tens of logical nodes. Examples are circuit breaker (abbreviated “XCBR”) and distance protection (“PDIS”). Each logical node (as illustrated in Figure 32) is composed of several data that represent some application-specific meaning (see Annex A for an overview of logical nodes and data).

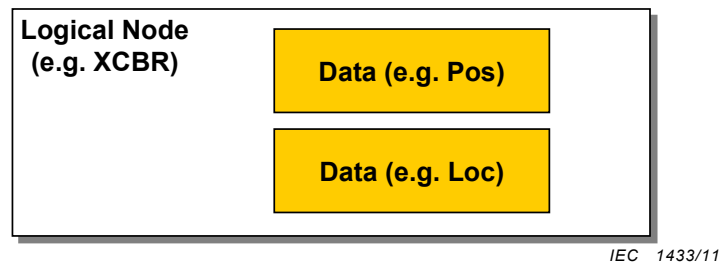


Figure 32 – Logical nodes and data (IEC 61850-7-2)

The data Pos as part of a circuit breaker is used to control the position and to report the status of the position; the data "Loc" stands for “local operation” and is a status indicating that the operation of the breaker from a remote location is enabled or not.

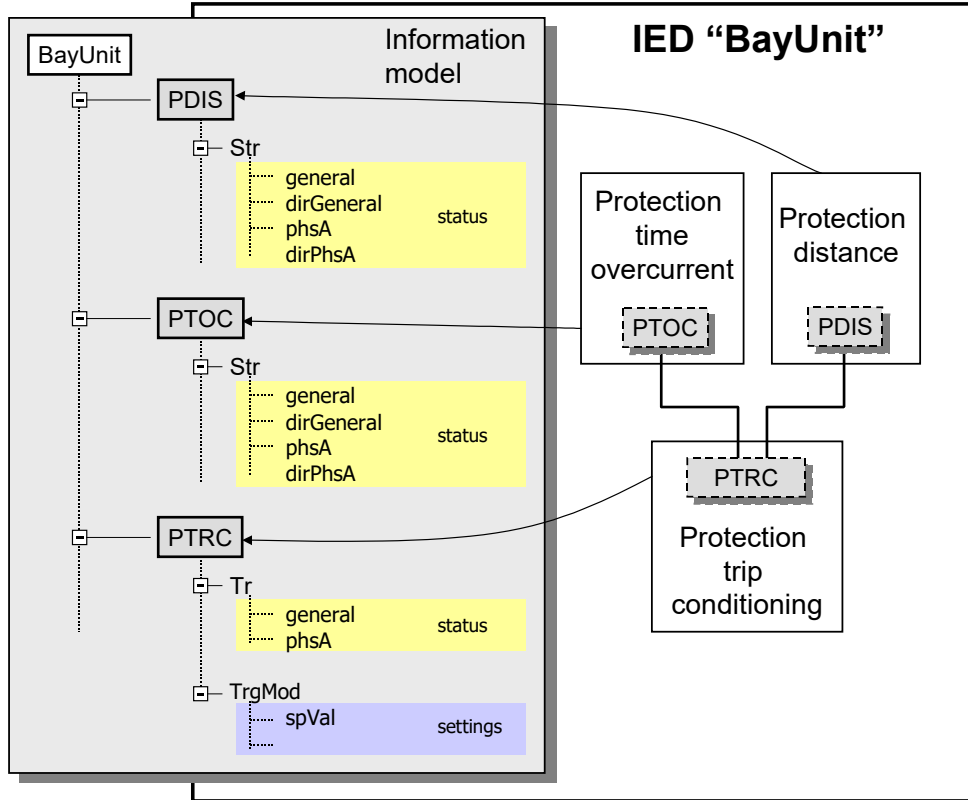
These data constitute the basis of most information exchanges over the network. Most interactions with a device are through data in logical nodes and services. What type of application information a specific data represents is defined in IEC 61850-7-3 (common data class), for example, double point status or measured value. Each common data class has services assigned to it that define the possible services that are allowed to be operated on this data. Some information may be writeable and readable while other information may be readable only. The so-called functional constraint (FC) defines this characteristic for each information of a specific data class. The information of a data is defined to be mandatory or optional. All services (for example, GetDataValues, Operate) are defined in IEC 61850-7-2.

The logical node names (for example XCBR for circuit breaker) and the names of data (for example Pos for the position of the real switch) define the standardised meaning (semantic) of the substation device. These abbreviated terms are standardised names that are used for communication (independent of the communication system used). The information model comprises many logical nodes, data, and data attributes.

This model is also used as a basis for the already mentioned substation configuration language (SCL) according to IEC 61850-6. The substation configuration describes which of the optional information is used in a specific device, what the instance names of all logical nodes are, what communication links exist, what the relation of the IEDs to the single line diagram is, and all information which is needed for the system engineering. The instance inherits everything from its class and assigns a unique name to it.

This standard makes use of a hierarchical organisation of data. Figure 33 shows an example of a real device “BayUnit”, protection function functions such as “Distance protection” (PDIS) and “Time Overcurrent” (PTOC) and “Trip conditioning” (PTRC). The process data of these basic functions, the basic functions and other important aspects of the bay unit are modelled

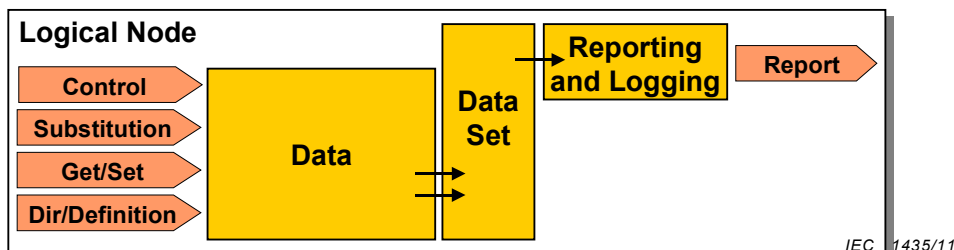
as data in a tree-structure. Each element of this tree is a data: the data at the top level is “Bay Unit” which contains “Distance protection”, “Time Overcurrent” and “Trip conditioning”. The “Distance protection” contains, for example, the data “Start” (Str) with different attributes such as “general” (general) and “Phase A” (phsA).



IEC 1434/11

Figure 33 – Simple example of modelling

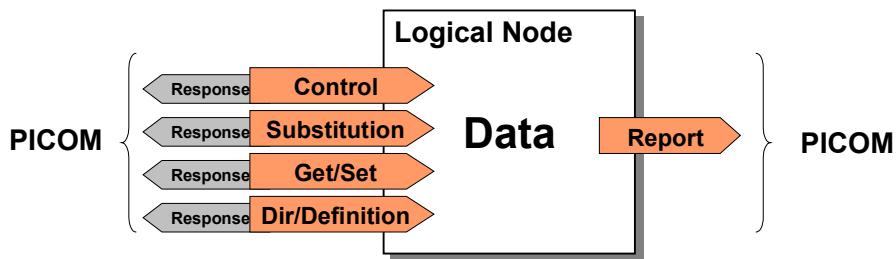
The elements of a logical node are illustrated in Figure 34. A control service represents the ability to control something in a device. This is modelled as data. To reset, for example, all LEDs in a device, only the value of the data "LEDs" has to be set to True. Data can be grouped into data sets and be reported immediately or logged for later query.



IEC 1435/11

Figure 34 – Basic building blocks

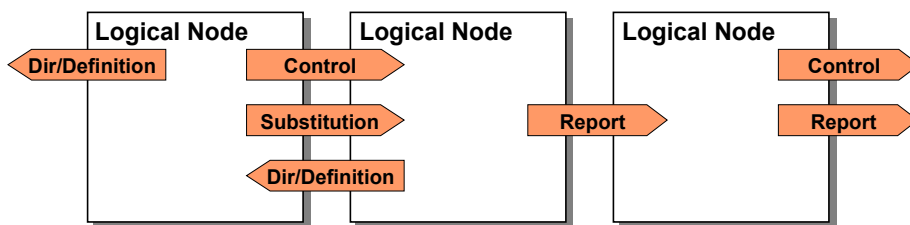
Control and report build one part of the interface of a logical node. Other services that operate on data are: substitution, for replacing a data value with a fixed one; get and set, for reading and writing data and data set values; Dir and Definition (GetDataDirectory, GetDataDefinition) retrieve the directory information of a data instance and the definition of a data instance. From an abstract viewpoint, the interfaces of a logical node can be summarised as illustrated in Figure 35. The services can be understood as carrying the information defined by PICOMS (Pieces of communication) as introduced in IEC 61850-5.



IEC 1436/11

Figure 35 – Logical nodes and PICOM

Logical nodes and data (contained in logical nodes) are the fundamental concepts that are used to describe real systems and their functions. Logical nodes function mostly as a container for data and can be placed anywhere in an IED. Each data defined in IEC 61850-7-4 has a specific meaning assigned to it. The data interact with their environment through their services. The concepts of logical nodes and data in the IEC 61850-7-x series defines the information that can be accessed in a logical node. The device that issues for example the request to retrieve data from a logical node can be modelled as a logical node, too. The information flow can be viewed between logical nodes (see Figure 36 and 9.5).



IEC 1437/11

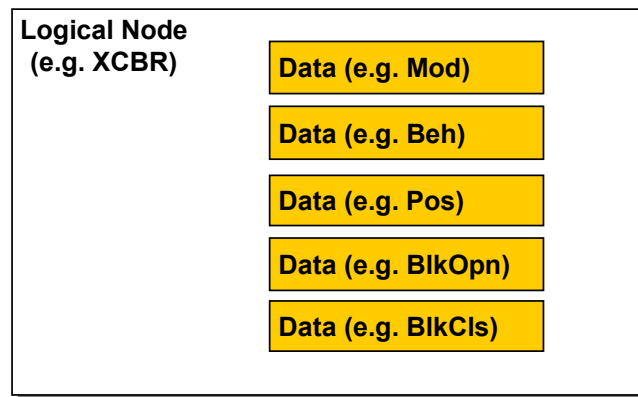
Figure 36 – Logical nodes connected (outside view in IEC 61850-7-x series)

From this point of view, the information flow is abstracted from any communication related information, for example, request/response notation.

Further building blocks and their services are explained in Clause 8.

Domain experts, for example of switchgear devices or power transformers, should primarily read and understand the logical nodes for switchgear devices (XCBR, XSWI, etc.) or for power transformers (YPTR, YLTC, etc.) and the data that belong to these logical nodes as defined in IEC 61850-7-4. IEC 61850-7-3 needs to be understood to see all the detailed information that may be exchanged with a device.

7.3 Mode and behaviour of a logical node



IEC 1438/11

Figure 37 – Mode and behaviour data (IEC 61850-7-4)

Mod and Beh are data which carry specific meanings in the context of this standard. In Figure 37, Mod is a controllable data representing the current operation mode of the circuit breaker logical node (on, blocked, test, test/blocked, off).

Switching the Mod to different states has an impact at the application level. However, the change of the Mod value has no effect on the communication related functions like reporting and GOOSE messages. The behaviour of those functions shall be handled by the appropriate services defined in IEC 61850-7-2.

Beh is a non-controllable data and has the same values as Mod. It represents the combination status of the mode of a logical node and the mode of the containing logical device (see 8.2).

The main requirements with regard to the different states of Mod/Beh (e.g. test) are defined in IEC 61850-5. The detailed requirements and application examples can be found in IEC 61850-7-4.

7.4 Use of measurement ranges and alarms for supervision functions

The logical nodes from the S group, Supervision, may be used to monitor physical parameters of a primary device. Each of these logical nodes has the following data:

- one or several measurements (usually from the common data class MV) of the physical parameters,
- one or several alarms related to the measured process.

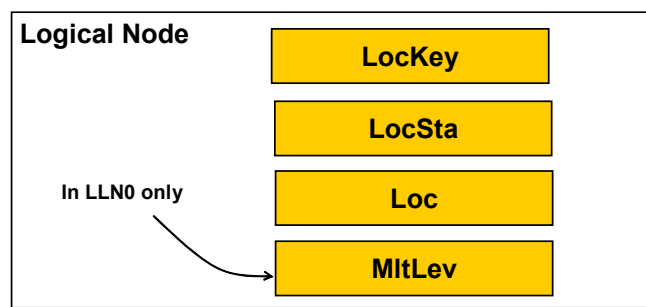
The measurement values are used to monitor a trend. The range of values (see 6.4.3.2) gives an indication of the limits the values are crossing. These limits only have a generic meaning as their values “high, low, high-high...” may have different meanings depending upon the type of measurement. For example, “high” may mean “take an action” or just “wait and see”.

On the other hand, the alarms give immediate information as to when some fast changes occur. Moreover, these alarms have specific names, like InsTr (insulation trip for device isolation), in order to standardize the semantic meaning of process variable supervision.

7.5 Data used for limiting the access to control actions

Control actions may be performed by an operator at different functional levels in the substation: station level, bay level and process level. As described in IEC 61850-5, the logical nodes including those involved in control actions may be allocated to different functional levels.

In order to limit the access to control actions at any level inside the substation and also from outside the substation, special data shall be used in controllable logical nodes (see Figure 38). For example, a breaker function (logical node XCBR) shall contain the mandatory data Loc indicating that the control authority for this function has been switched from remote to local by certain means. The logical node XCBR may also contain the data LocSta indicating that the control authority has been switched from the station level to an upper level, e.g. the NCC level. The data LockKey is intended to be used to signal the state of a physical local/remote key or toggle switch. As an example, the position of a key switching simultaneously several logical nodes in local mode, e.g. at the entire IED level, will be indicated by the data LockKey in the logical node LLN0. MltLev is a setting which allows for the choice of different modes of switching authority. MltLev is intended to be applicable to a whole logical device (see 8.2.1) and therefore shall only be present in the logical node LLN0. The details concerning the values of MltLev and the related checking conditions a real device shall perform when receiving a control request can be found in IEC 61850-7-4.



IEC 1439/11

Figure 38 – Data used for limiting the access to control actions (IEC 61850-7-4)

7.6 Data used for blocking functions described by logical nodes

A specialized LN input is described by the data BlkRef defined in the common logical node class (see IEC 61850-7-4). BlkRef can be used to reference an incoming blocking signal. A typical application is the dynamic blocking of the operation of a protection function or a recloser function. BlkRef is associated with the data Blk intended to inform about the blocked status of the function represented by the concerned LN.

7.7 Data used for logical node inputs/outputs blocking (operational blocking)

7.7.1 General

It may be required by an operator to manually block inputs/outputs of a function. For example, it shall be possible to block incoming signals having an effect on control actions or on process output activation. As shown on the left side of Figure 39, some of those blockings are realized by the use of control services (see IEC 61850-7-2) and dedicated data defined in the common logical node class (see IEC 61850-7-4). Other blocking mechanisms use the “Set” service and dedicated data attributes as shown on the right side of the figure.

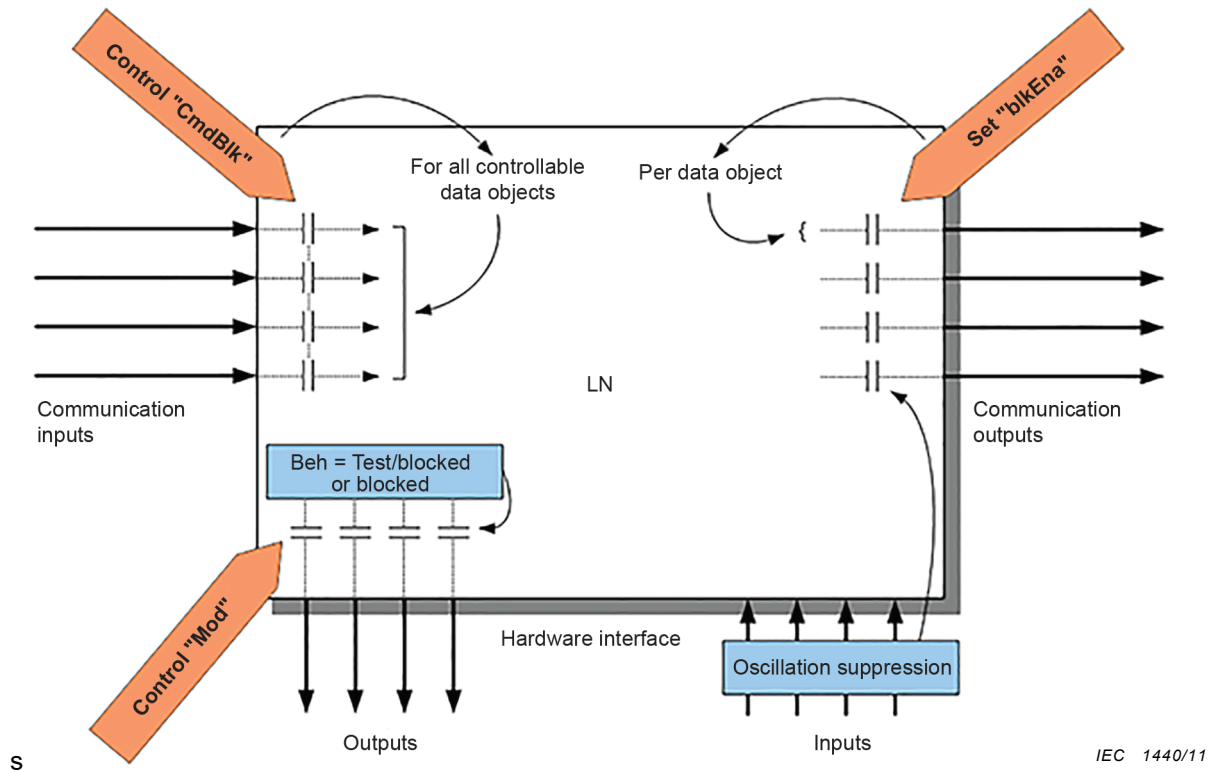


Figure 39 – Data used for logical node inputs/outputs blocking (IEC 61850-7-4)

7.7.2 Blocking incoming commands

The data CmdBlk (command blocked) is used to block incoming commands at LN level. It shall be applied to controllable data of any LN, i.e. to the common data classes SPC, DPC, INC, etc. When the blocking is activated, the command shall not be accepted and an appropriate control negative response shall be sent to the client who initiated the command.

The data CmdBlk can also block commands coming in via GOOSE messages (e.g. the data OpOpn/OpCls in the LN CSWI), thus disabling any action which could activate the outputs to the process.

The data BlkOpn/BlkCls in the LN XCBR are similar to the data CmdBlk but have a more precise semantic attached to them.

The data CmdBlk shall have no effect on the controllable data Mod or CmdBlk itself.

7.7.3 Blocking process outputs

The data Mod (Mode) is used to put an LN in different modes. The mode blocked (Beh= test/blocked or Beh=blocked) shall be interpreted as the mode which explicitly disables any physical outputs having an effect on the process. For example, when XCBR.Beh = test/blocked or blocked, the physical outputs for closing/opening the breaker do not operate and the breaker does not close or open respectively.

7.7.4 Blocking the communication of status outputs updates

An operator, or an automatic function, may block the update of the value of communication outputs (e.g. updates related to oscillating inputs). The blocking of the update of the value shall be available at data level and thus is realized by the data attribute blkEna (blocking enabled). When the data attribute is set, as a result, the quality of the associated data becomes operatorBlocked and oldData (see IEC 61850-7-2).

7.8 Data used for testing

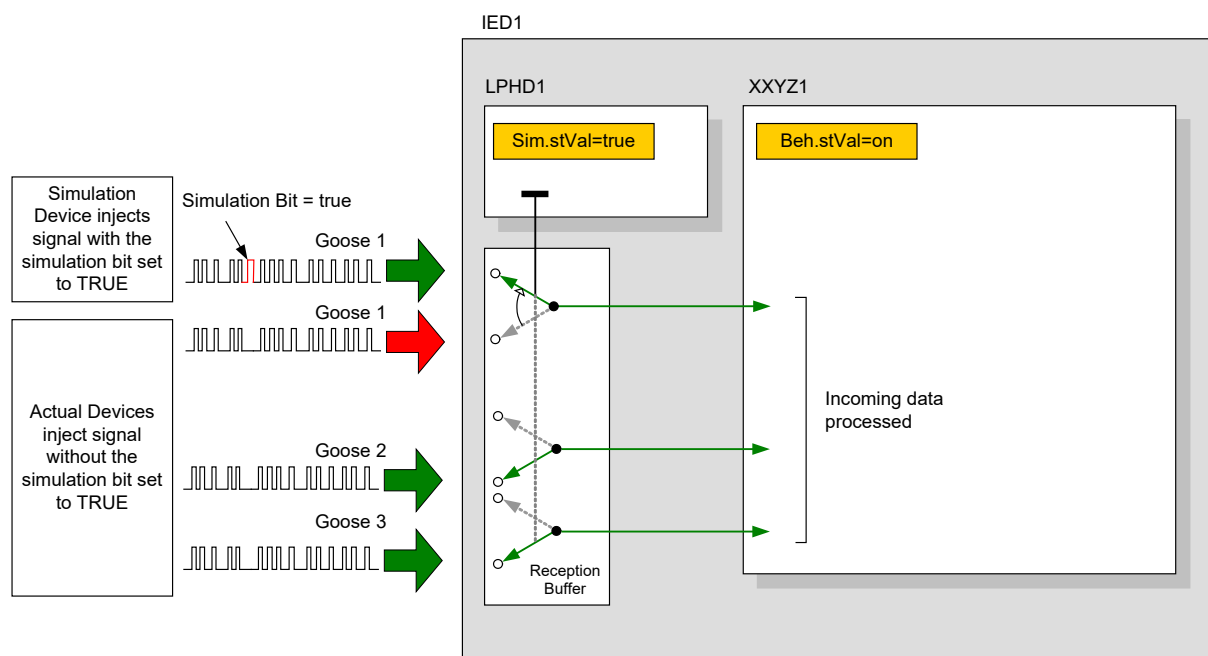
7.8.1 General

In order to carry out functional, commissioning or maintenance tests, a communications network-based SAS that supports testing functions should offer some of the following facilities:

- at the IED level, the option of receiving multicast simulation signals instead of actual signals (see 7.8.2);
- at the LN (function) level, the option of receiving test input signals instead of actual signals (see 7.8.3);
- at the LN (function) level, the option of setting a function or a group of functions of the system in test mode (see 7.8.4).

7.8.2 Multicast signals used for simulation

Figure 40 shows an IED (IED1) receiving simultaneously simulation and actual signals.



IEC 1441/11

Figure 40 – Data used for receiving simulation signals

On the left side of the figure, actual devices are sending three GOOSE messages (GOOSE 1, GOOSE 2 and GOOSE 3) to the function represented by the LN XXYZ1 in the IED1. Additionally, a simulation device is sending the same GOOSE 1 message, but this time with the simulation bit being set to TRUE.

To allow the IED1 to process the simulated GOOSE1 message instead of the actual GOOSE 1 message, the data `Sim.stVal` in the LN LPHD1 shall be set to TRUE. As of this moment, the simulated signal shall replace the actual signal. This will remain as such until `Sim.stVal` is set back to 'FALSE'. During this time, the other signals, GOOSE 2 and GOOSE 3 are normally processed by the IED1. Note that the switching between a normal signal and a simulated signal shall be done for the entire IED. In the example, this means that if a new simulated GOOSE 2 signal appears, it will replace the actual GOOSE 2 signal. The preceding example does not imply any specific sequence with regards to connecting the simulation device and setting the data `Sim.stVal` to TRUE. Different sequences may be used. However, a state machine diagram can be found in IEC 61850-7-4 that describes the different states involved and illustrates the supervision of GOOSE/SV subscription with regards to `LGOS/LSVS.St` and `LGOS/LSVS.SimSt`.

The example in Figure 40 not only applies to GOOSE messages but to sampled values (SMV) as well.

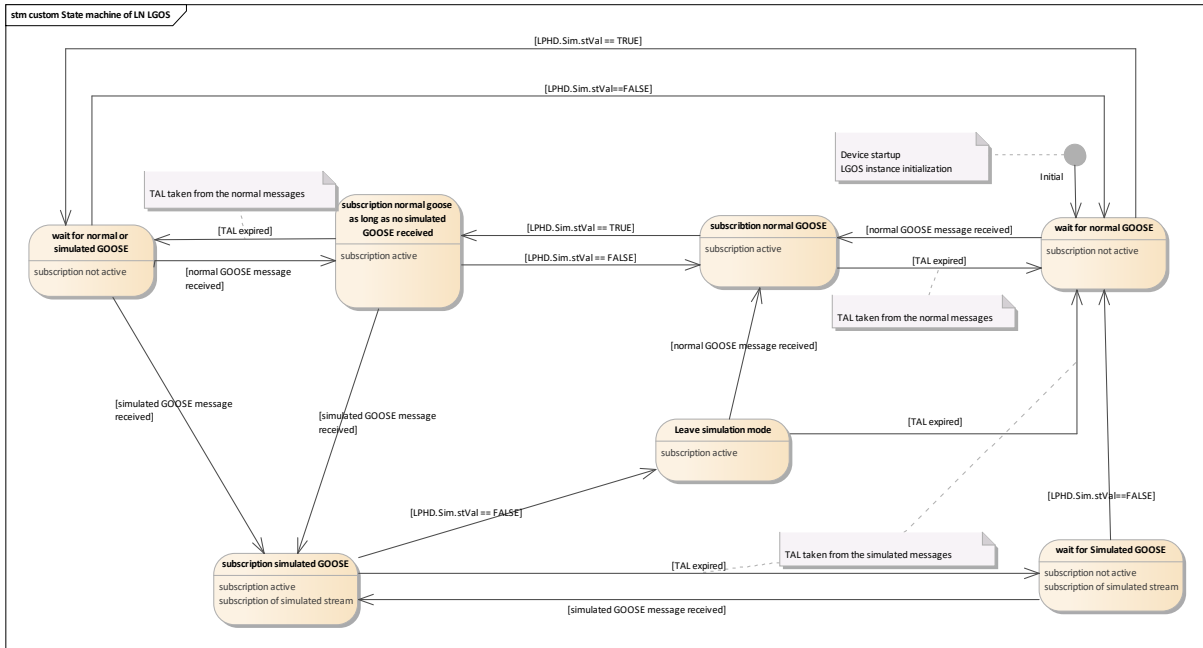


Figure 80 – GOOSE subscription supervision state machine

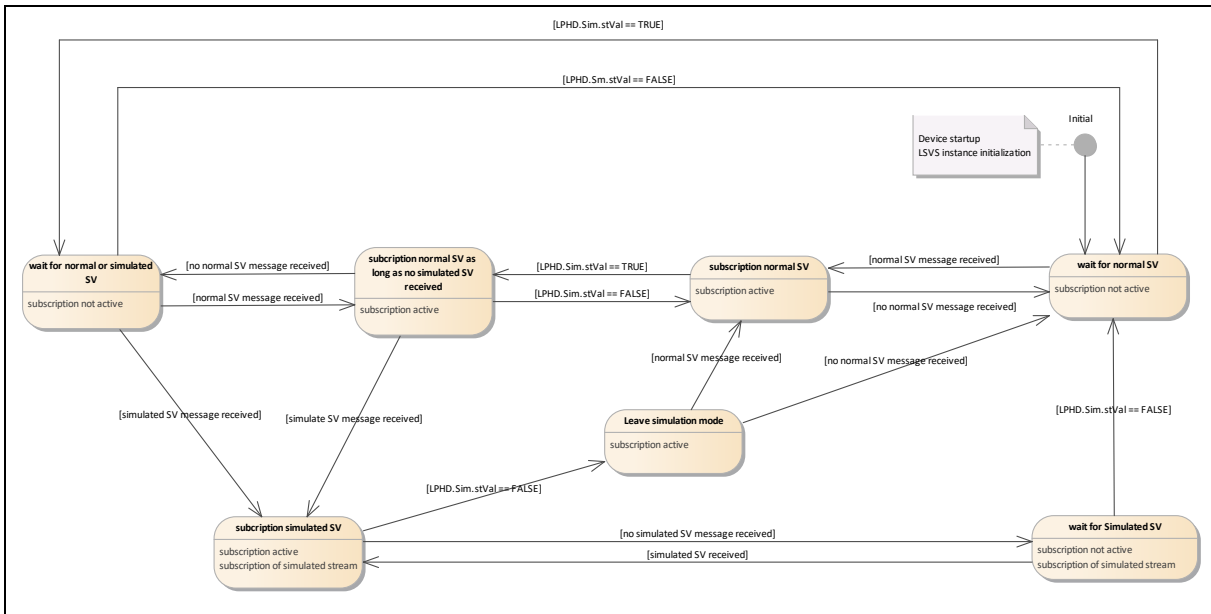


Figure 81 – SV subscription supervision state machine

7.8.3 Input signals used for testing

Each LN may have one or several data named InRef1 to InRefn and used as input signals to the function represented by the LN (see 9.6). The common data class of InRef1 is ORG, “object reference setting group”. The data can be used to switch between actual and test incoming signals as shown in the example of Figure 41.

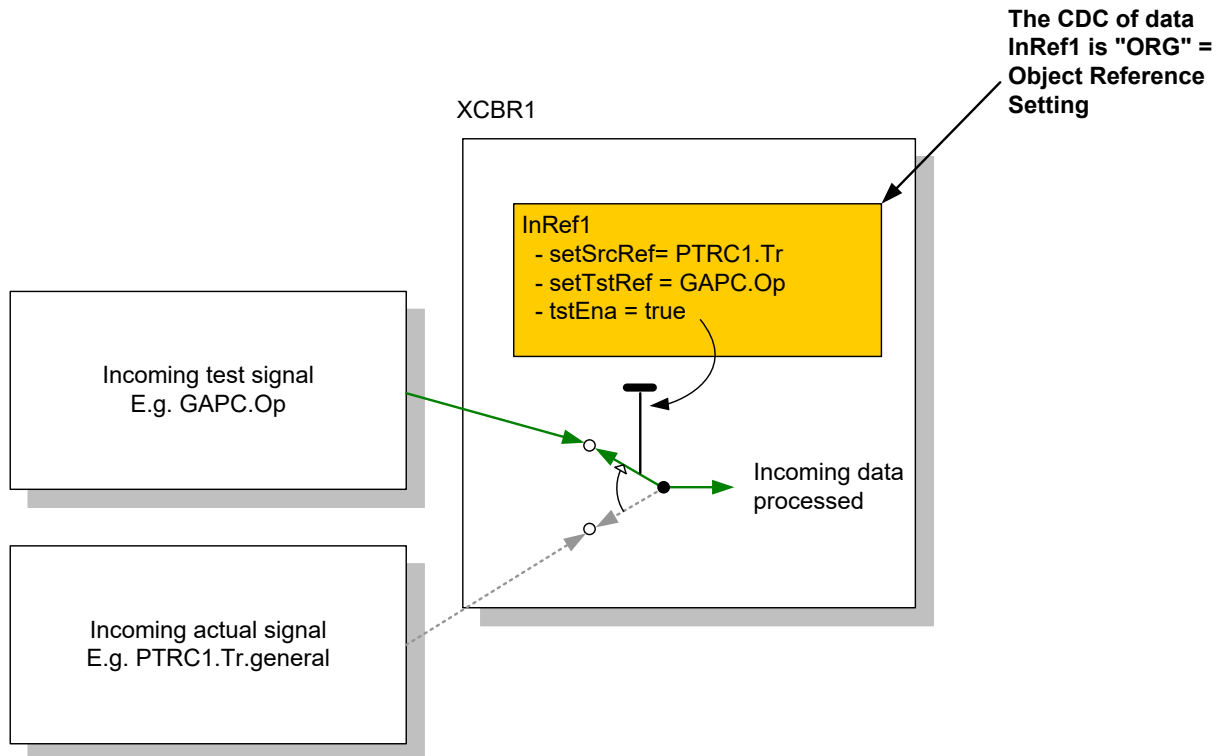


Figure 41 – Example of input signals used for testing

In normal operation, the breaker function (XCBR1) receives as an input the signal PTRC1.Tr.general intended to trip the breaker. For functional testing, this signal can be changed for another incoming signal, GGIO1.Ind1.stVal in the example. This change is made by setting the value of the attribute InRef1.tstEna to TRUE. The common data class ORG also has the attributes setSrcRef and setTstRef used to indicate the origin of the incoming actual and test signals. The values of the attributes setSrcRef and setTstRef shall also contain the logical device names of the incoming signals. In the example, they have intentionally not been specified as the concept of the logical device is introduced further in this document. The signal specified in setTstRef, the associated control block srcTstCB shall be consistent and relate to a fully configured (i.e. including the communication configuration when necessary) control block in the SCD file.

7.8.4 Test mode

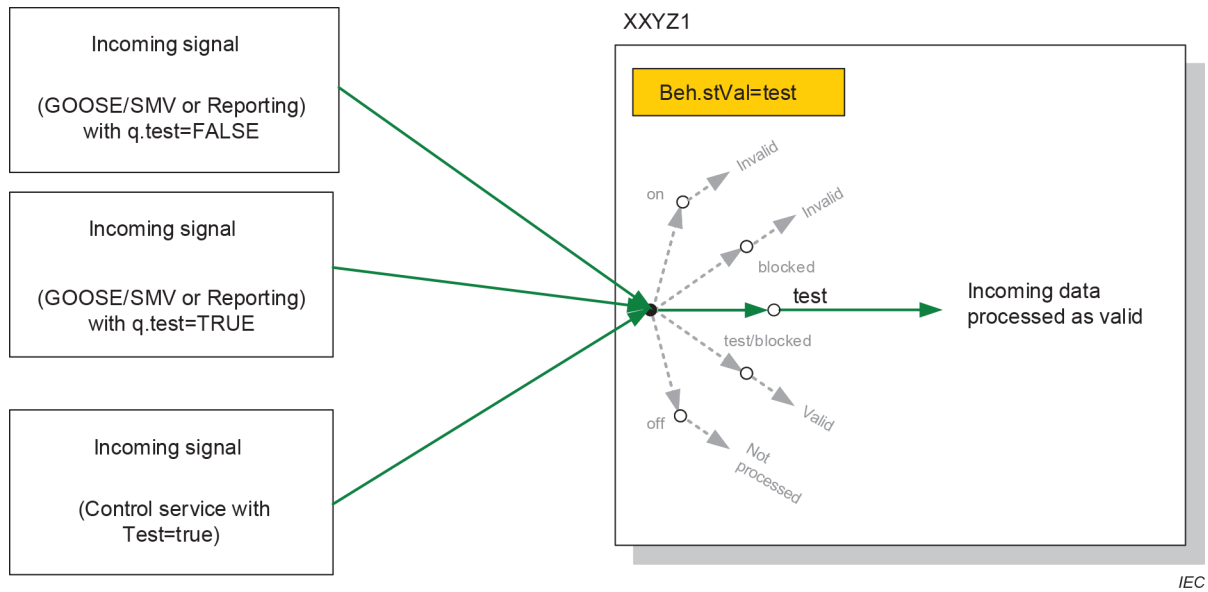


Figure 42 – Test mode example

Figure 42 shows an example of a function (LN XXYZ1) in test mode. In order to make the function accept an incoming signal as valid for testing purposes, the following conditions shall be observed:

- the mode of the function ($Beh.stval$) being part of the test shall be set to 'test' or 'test/blocked',
- in case of GOOSE or reporting services, the quality value of the incoming data shall be 'test',
- in case of control services, the control service parameter $Test$ value shall be TRUE.

The detailed requirements for the different states of Mod/Beh can be found in IEC 61850-7-4.

7.9 Logical node used for extended logging functions

Some applications require the recording of sequential events whose events are not only defined by the data that triggered the event but also by other useful information that needs to be captured simultaneously when the event occurred. The function used to set the additional data to be captured is represented by the LN GLOG: Generic Log, as shown in Figure 43.

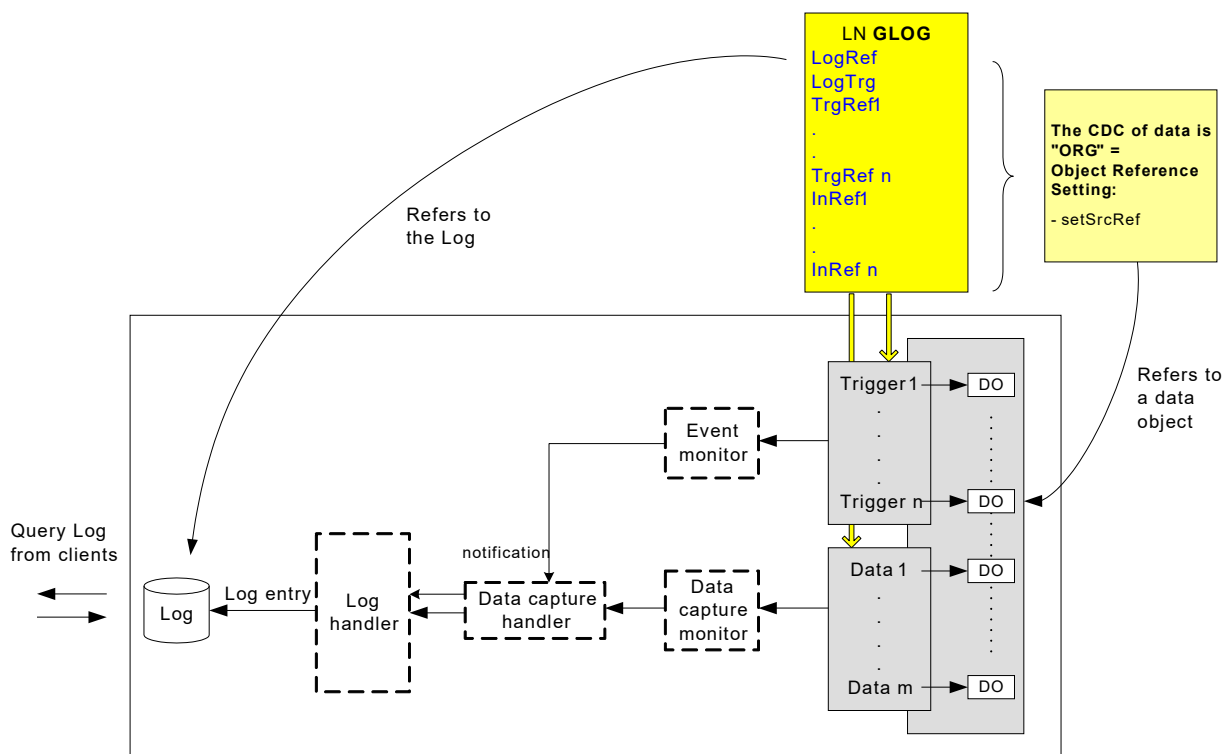


Figure 43 – Logical node used for extended logging functions (GLOG)

The LN GLOG is associated to a Log (see 6.4.3.3.3) and contains settings (CDC ORG) which are used to reference event triggers and the additional data to be captured. In Figure 43, it can be seen that the LN GLOG contains the information about the triggers (TrgRef1 to TrgRefn) and the additional data (InRef1 to InRefn).

Once an event is detected and notified, the captured data is added to the Log. The reason for inclusion, as defined in IEC 61850-7-2, shall be application-trigger.

GLOG only represents additional data to be logged when a trigger occurs. The usage of GLOG is application specific and therefore doesn't need to be controlled by a LCB (see 6.4.3.3.3).

If needed, the triggers themselves may be included into the log by two means:

- the use of a log control block and a dataset which includes the trigger references;
- by fixing the settings InRefx equal to the trigger references.

There shall be only one Log per LN GLOG.

8 Device view

8.1 General

Real devices host mainly:

- logical nodes and data – representing the real application functions and associated information visible from the communication network (data defined in IEC 61850-7-4),
- information about real devices – representing information about the resources of the host itself and (if applicable) about the real equipment connected to the host device (specific logical nodes and data defined in IEC 61850-7-4),

- communication services and mapping to specific communication systems – representing the supported information exchange services (defined in IEC 61850-7-2 and SCsMs).

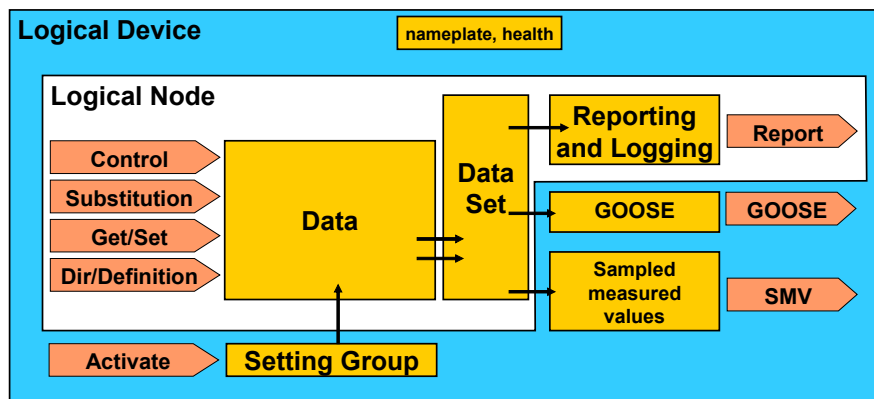
The second and third items require further components in the model. To define information about the devices and to model those communication aspects that are applicable to more than one logical node requires a model that comprises the logical nodes and the further information and service models.

8.2 Second modelling step – logical device model

8.2.1 The logical device concept

For communication purposes (beyond a logical node), the concept of a logical device has been introduced. A logical device is mainly a composition of logical nodes and additional services (for example GOOSE, sampled value exchange, setting groups) as illustrated in Figure 44. The grouping of logical nodes in logical devices is based on common features of these logical nodes. For example, the modes of all these nodes are normally switched on and off together, or in the test mode.

NOTE GOOSE is used to very rapidly exchange input and output data mainly of relays.



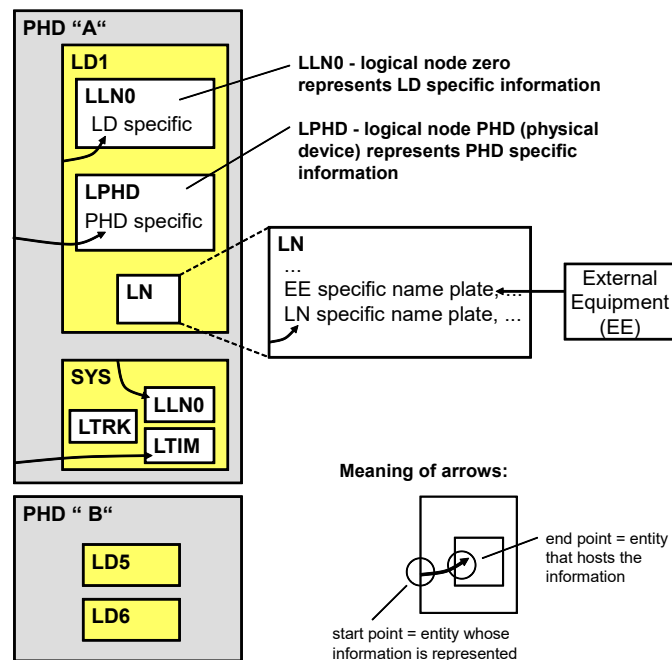
IEC 1445/11

Figure 44 – Logical device building block

Logical devices provide information about the physical devices they use as host (nameplate and health) or about external devices that are controlled by the logical device (external equipment nameplate and health). Logical devices reside in physical devices as shown in the example in Figure 45. Only those aspects of physical devices that are defined as visible to the network are of interest in this standard.

In the example of Figure 45, logical device “LD1” contains three logical nodes: LLN0, LPHD and LN, the latter representing an application specific LN as defined in the IEC 61850-7-x series. The logical node zero (LLN0) represents common data of the logical device. For example, the mode of LLN0 is used to control the mode of the whole logical device and, as a result, the mode of every logical node being part of the logical device. The logical node physical device (LPHD) represents common data of the physical device hosting the logical device. LLN0 shall be defined in every logical device while LPHD shall be defined in at least one logical device. There exist many other system logical nodes (Group L) belonging to the IED, e.g. LTRK which allows tracking of service parameters for a server within the IED or LTIM which serves for diverse configurations regarding the local time of an IED. Figure 45 shows an example of a good practice. The logical device called “SYS” groups all system logical nodes belonging to the same IED. Note that these system logical nodes could also be spread in a hierarchy of logical devices (see 8.2.5).

On the right-hand side, the representation of for example nameplate information about the primary equipment is defined as data of the logical node that represents the primary equipment.



IEC 1446/11

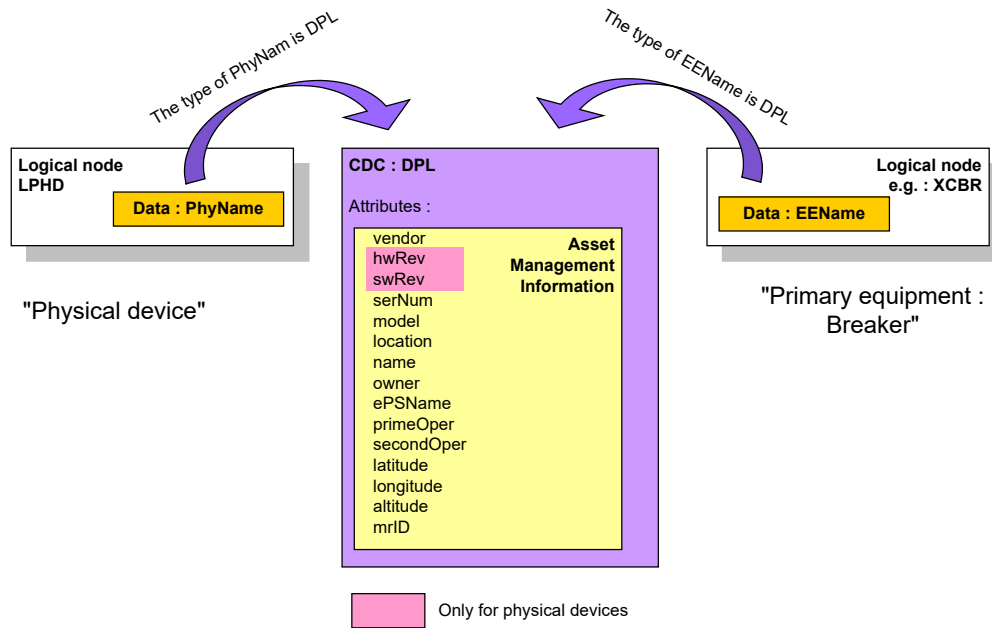
Figure 45 – Logical devices and LLN0/LPHD

The functional content of logical devices, e.g. logical nodes, is not standardized by the series of documents IEC 61850-7-x, neither the name of logical devices.

8.2.2 The device nameplate

Physical devices and affiliated primary equipment are identified by the device nameplate information. This information comprises asset management related data like the serial number, the location, the owner, the electric power system to which the primary equipment is connected, the operator of the device, etc. The device nameplate information is part of the common data class DPL in IEC 61850-7-3. The DPL attributes apply to physical devices and to primary equipment as well. Of course, some attributes only make sense when they are used for physical devices.

Figure 46 shows the DPL common data class used in two different logical nodes. The data PhyNam in logical LPHD represent the nameplate of a physical device while the data EENam is the nameplate of primary equipment represented by the logical node XCBR.



IEC 1447/11

Figure 46 – The common data class DPL

8.2.3 Gateways and proxies

Gateways are network interconnection devices that translate protocols to other protocols. For example, gateways may convert non IEC 61850 data into IEC 61850 data.

Proxies are special devices that mirror logical devices located in other IEC 61850 physical devices. Hence, these logical devices are, from a functional point of view, transparent. They can be identified independently of their location (in a separate device connected to the network or in a proxy device).

Logical devices allows for the building of proxies and/or gateways.

Figure 47 shows how a physical device is mapped to a device acting as a proxy and/or a gateway. In the figure, LN refers to an application specific LN as defined in the IEC 61850-7-x series.

The logical device IED1LD1 is “copied” to the proxy/gateway. The LPHD of IED1LD1 in the proxy/gateway represents the physical device IED1.

Logical devices that mirror logical devices of other physical devices shall provide a LPHD that represents the remote physical device on which the original LD resides (e.g., A_LD1). These logical devices shall have the data LPHD.Proxy.stVal of the LPHD set to TRUE.

Logical devices that do not mirror logical devices of other physical devices shall provide a LPHD that represents the physical device on which they reside. In Figure 47, this logical device is PXY_GTW_LD1, which shall be implemented to represent the information about the proxy/gateway itself. The logical nodes LLN0 and LPHD of this logical device shall represent information about the proxy/gateway device. The logical device may also contain domain specific logical nodes.

In the example, the physical device IED2 is also proxied by the proxy/gateway. The device contains a hierarchy of logical devices. In this case, the LPHD that represents the remote physical device in the proxy/gateway is declared only at the root of the hierarchy.

Figure 47 also shows how non IEC 61850 external physical devices are mapped to the proxy/gateway device. IED3LD1 is used to represent the information related to the non IEC 61850 physical device IED3. The LPHD of IED3LD1 represents the physical device IED3. The data supplied by IED3 is mapped to different logical nodes into IED3LD1.

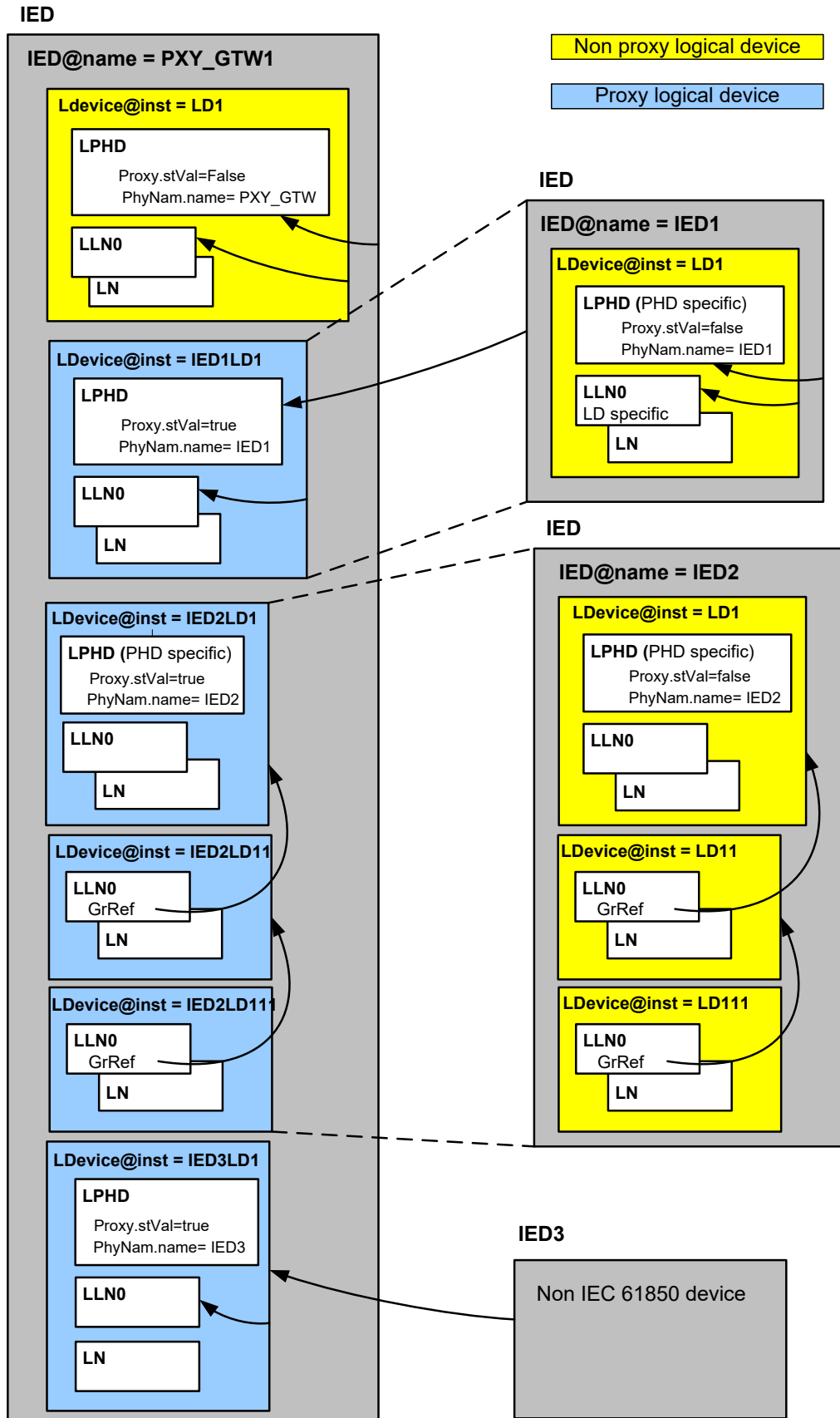


Figure 47 – Logical devices in proxies or gateways

Figure 79 shows the same example using the functional naming of the logical devices. One can see that it is possible to make the names of the proxied logical devices exactly the same as the ones in the source logical devices. This is possible as long as the respective access points (Server of the Proxy and original Server) are not connected to the same network: the name of the logical device shall be unique within the scope of a network.

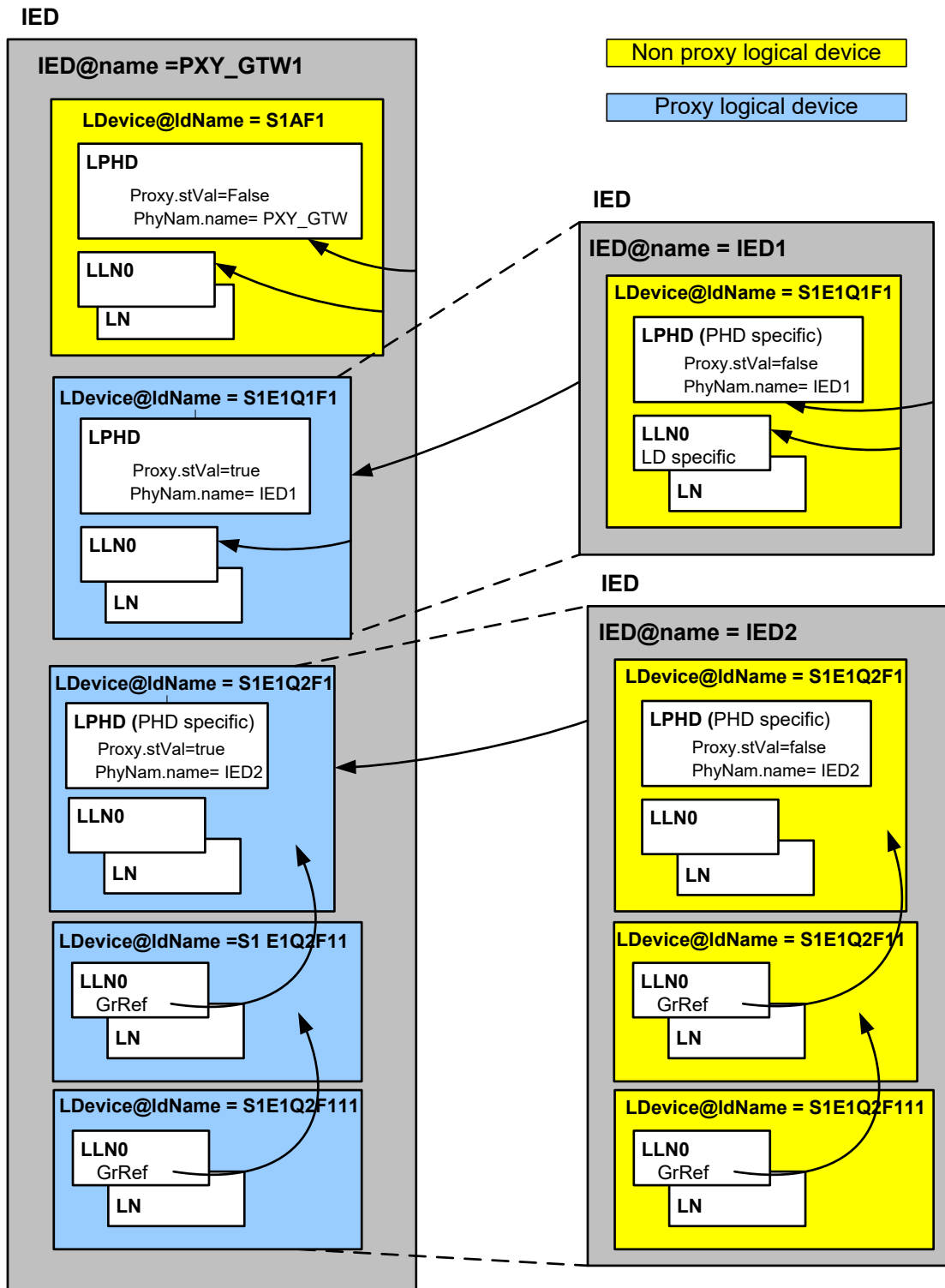


Figure 79 – Logical devices in proxies or gateways (functional naming)⁴

⁴ IED@name, LDevice@inst and LDevice@IdName are SCL elements (IEC 61850-6).

8.2.4 Logical devices for monitoring external device health

The logical device concept may be used to supervise the health of an external device. Figure 48 shows an example where the gateway approach is reused. PHD "B" is a measuring device connected to a sensor and communicating with PHD "A" using a non-IEC 61850 means of communication. The T node is used to represent the sensor health while the LPHD node is used to represent the health of the external device.

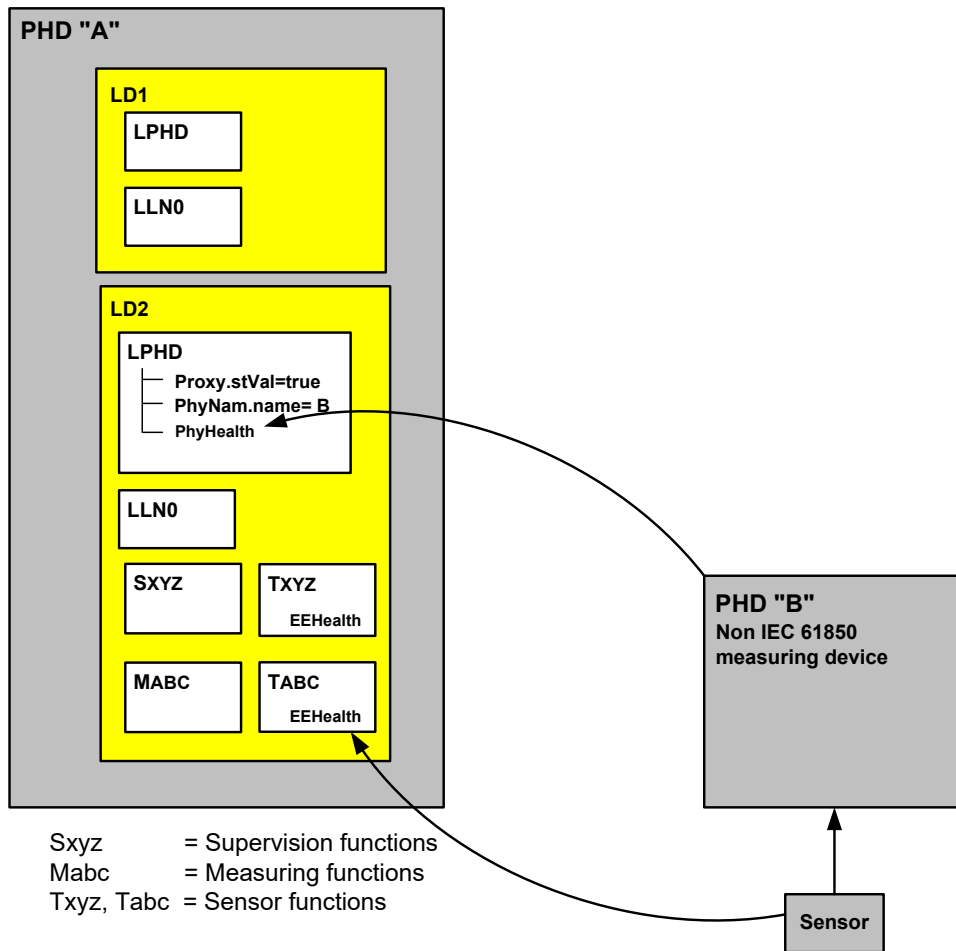
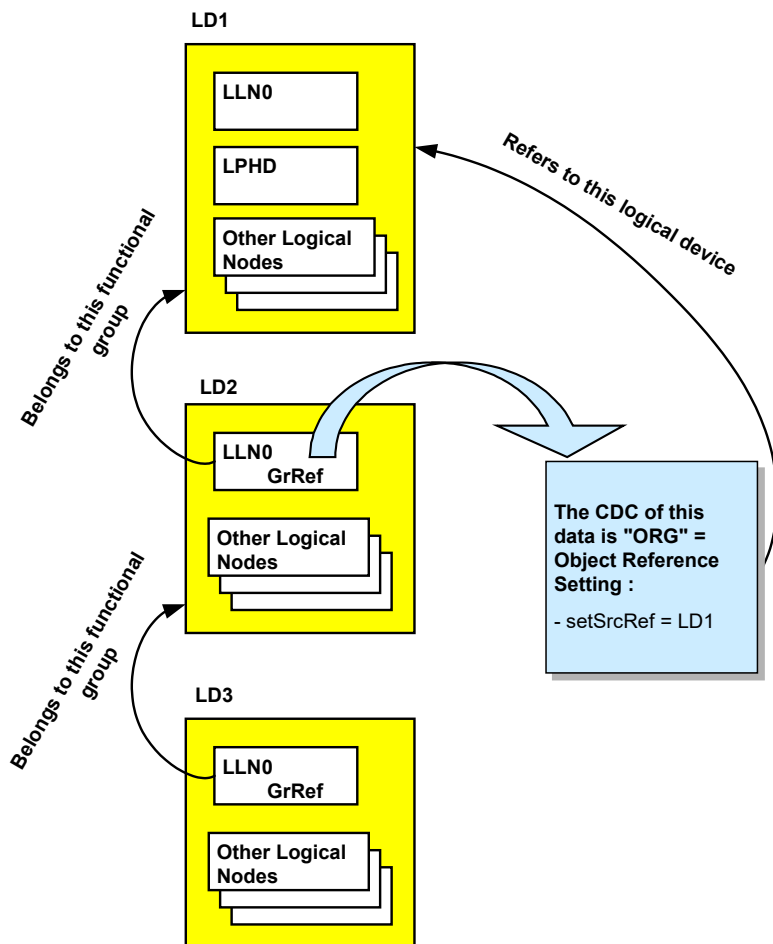


Figure 48 – Logical devices for monitoring external device health

8.2.5 Logical devices management hierarchy

Logical devices are used to represent a group of typical automation, protection or other functions. The functions are defined as logical nodes contained and managed in logical devices. This simple hierarchy may not be sufficient to model complex functions, e.g. distance protection. These functions are usually made of nested functions and sub-functions, represented by logical nodes, which shall be grouped together according to a management hierarchy. The hierarchy determines how the mode (e.g. On, Off, Test, ...) or the health of these functions and sub-functions is managed. The concept is shown in Figure 49 where LD1 is called the "root" logical device.



IEC 1450/11

Figure 49 – Logical devices management hierarchy

LLN0 of the logical device LD2 contains a setting data named GrRef whose common data class is ORG, “object reference setting”. The referenced value of GrRef is LD1, meaning that the logical device function refers to the functional group represented by the logical device LD1. Likewise, LLN0 of LD3 refers to the functional group represented by the logical device LD2. In other words, LNs in LD3 are sub-functions of LD2.

In Figure 49, the mode of the LNs in LD3 may be changed individually or globally by means of LLN0 of LD3. Their mode may also be changed either by means of LLN0 of LD2 or by means of LLN0 of LD1. For example, if the mode of the functional group LD2 is set to “Off”, it not only set the behaviour of all logical nodes in LD2 to “Off” but also the behaviour of all logical nodes in LD3. Switching the mode of LD1 will affect the behaviour of all logical devices and logical nodes belonging to the functional group LD1, i.e. all logical nodes in LD1, LD2 and LD3.

The health (see Figure 63) of the various LN and LD is also managed within the hierarchy. In Figure 49, the health of LLN0 in LD1 reflects the worst value of “Health” of the logical nodes that are part of the logical devices LD2 and LD3.

The rules with regard to the concept of the logical device management hierarchy are the followings:

- a logical device management hierarchy shall only be defined in a single physical device,
- within a logical device management hierarchy, only the logical node class LLN0 can refer to another logical device,

- the logical node class LPHD shall be present in any root logical device, i.e. any LD where the LLN0 does not contain a data object GrRef. It may also be present at other levels of the hierarchy,
- a proxy LD shall only be present at the root logical device, applying to all other level of the associated hierarchy,
- the behaviour of any logical node or logical device in a branch of the logical device management hierarchy is a function of the modes along the branch from the leaf to the root logical device (see example above),
- if setting group control blocks are present, they shall appear only once in any branch of the logical device management hierarchy. The settings in a branch shall belong to the setting group control block of that branch.

9 Communication view

9.1 General

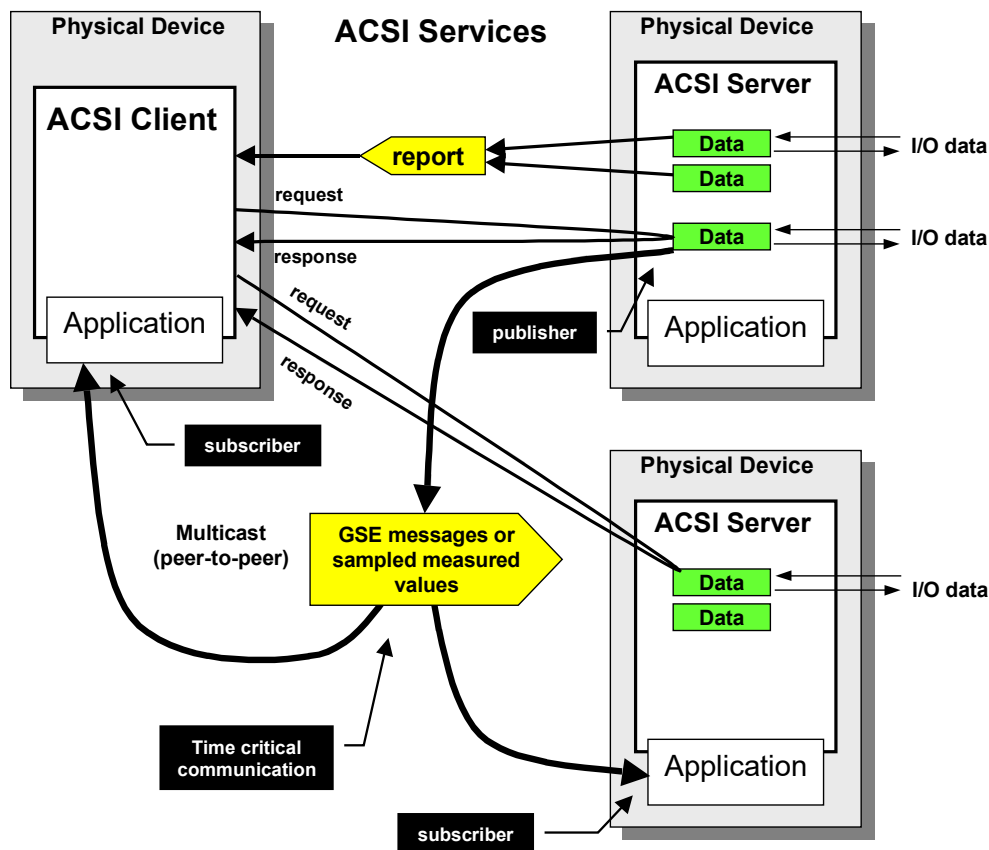
Communication systems abstract mainly from the application and device view. On the other side, the communication system, the devices, and the application are closely related. Clause 9 introduces the communication models. After that, the relations between these views are discussed.

9.2 The service models of the IEC 61850 series

The services are defined using an object-modelling technique. The service interface uses an abstract modelling method. Abstract means that the definition is focused on the description of what the services provide. The concrete messages (and their encoding) to be exchanged between devices (how the services are built) are not defined in this part of the standard. These concrete messages are specified in the specific communication service mappings (SCSMs in IEC 61850-8-x and IEC 61850-9-x series).

NOTE This abstraction allows various mappings appropriate for different requirements and following state-of-the-art in communication technology without in such a case changing the model and, therefore, databases, etc.

The ACSI (Abstract communication service interface) defines common utility services for substation devices. The two groups of communication services are depicted in Figure 50. One group uses a client-server model with services like control or get data values. A second group comprises a peer-to-peer model with GSE services (used for time-critical purposes, for example, fast and reliable transmission of data between protection IEDs, from one IED to many remote IEDs) and with the sampled value services for transmissions based on a periodic basis.



IEC 1451/11

Figure 50 – ACSI communication methods

Real clients and servers can be connected by a variety of communication systems. Communication media may have geographic and utilisation constraints, such as limited bit rates, proprietary data link layers, restricted times for use, and satellite hop delays. Systems may be hierarchical, with a few central sites authorising and managing the interactions with a large number of “field” sites, or it may be networked with peer-to-peer interactions. Communication media may have varying configurations, such as point to multi-point, multi-drop, meshed, hierarchical, WAN-to-LAN, intermediate nodes acting as routers, as gateways, or as data concentrator databases, etc.

Table 5 lists the ACSI service models and services.

Table 5 – ACSI models and services

Service model	Description	Services
Server	Represents the external visible behaviour of a device. All other ACSI models are part of the server.	GetServerDirectory
Application association	Provision of how two devices can be connected. Provides different views to a device: restricted access to the server's information and functions.	Associate Abort Release
Logical device	Represents a group of functions; each function is defined as a logical node.	GetLogicalDeviceDirectory
Logical node	Represents a specific function of the substation system, for example, overvoltage protection.	GetLogicalNodeDirectory GetAllDataValues
Data	Provides a means to specify typed information, for example, position of a switch with quality information, and timestamp.	GetDataValues SetDataValues GetDataDefinition GetDataDirectory
Data set	Allow to group various data together.	GetDataSetValues

Service model	Description	Services
		SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory
Setting group control	Defines how to switch from one set of setting values to another one and how to edit setting groups.	SelectActiveSG SelectEditSG SetEditSGValues ConfirmEditSGValues GetEditSGValues GetSGCBValues
Reporting and logging	Describes the conditions for generating reports and logs based on parameters set by the client. Reports may be triggered by changes of process data values (for example, state change or deadband) or by quality changes. Logs can be queried for later retrieval. Reports may be sent immediately or deferred (buffered). Reports provide change-of-state and sequence-of-events information exchange.	Buffered RCB: Report GetBRCBValues SetBRCBValues Unbuffered RCB: Report GetURCBValues SetURCBValues Log CB: GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues
Generic substation events (GSE)	Provides fast and reliable system-wide distribution of data; peer-to-peer exchange of IED binary status information. GOOSE means Generic object oriented substation event and supports the exchange of a wide range of possible common data organised by a DATA-SET	GOOSE CB: SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues
Transmission of sampled values	Fast and cyclic transfer of samples, for example, of instrument transformers.	Multicast SVC: SendMSVMessage GetMSVCBValues SetMSVCBValues GetMsvReference GetMSVElementNumber
Control	Describes the services to control, for example, devices or parameter setting groups.	Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate
Time and time synchronisation	Provides the time base for the device and system.	TimeSynchronization
File transfer	Defines the exchange of huge data blocks such as programs.	GetFile SetFile DeleteFile GetFileAttributeValues

9.3 The virtualisation

The ACSI provides access to the real data and real devices through a virtual image as depicted in Figure 51. A virtual image that represents the real data of devices is made visible and accessible through ACSI services. A computer may request services, for example, get data values, or may receive spontaneously reported values from the controller.

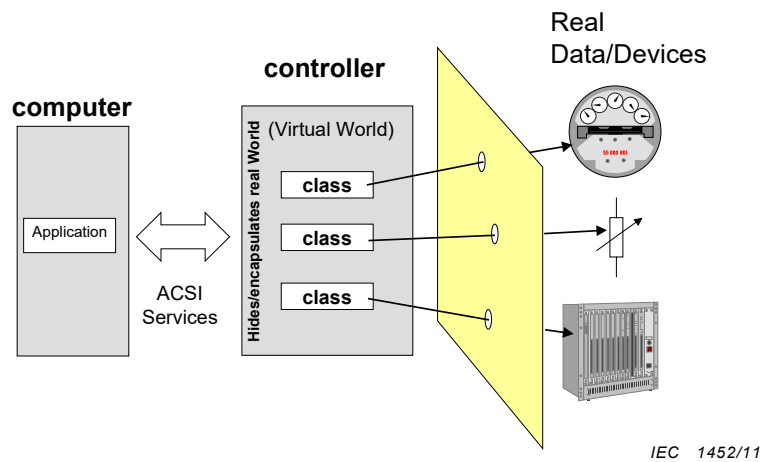


Figure 51 – Virtualisation

The virtual view can be used (as shown in Figure 52) to describe and represent the complete behaviour of a device. Any other device, another controller or even a SCADA system, a maintenance system, or an engineering system may use the ACSI services to inter-operate with that device. A service request received is independent of the device that has requested the service.

The communication system provides means to prevent that every single computer in the whole network is able to connect to any device and see and modify all information of any device. There are diverse access schemes defined that restrict the “visibility” of a device or particular data of a device. An operator may for example not be allowed to change protection settings.

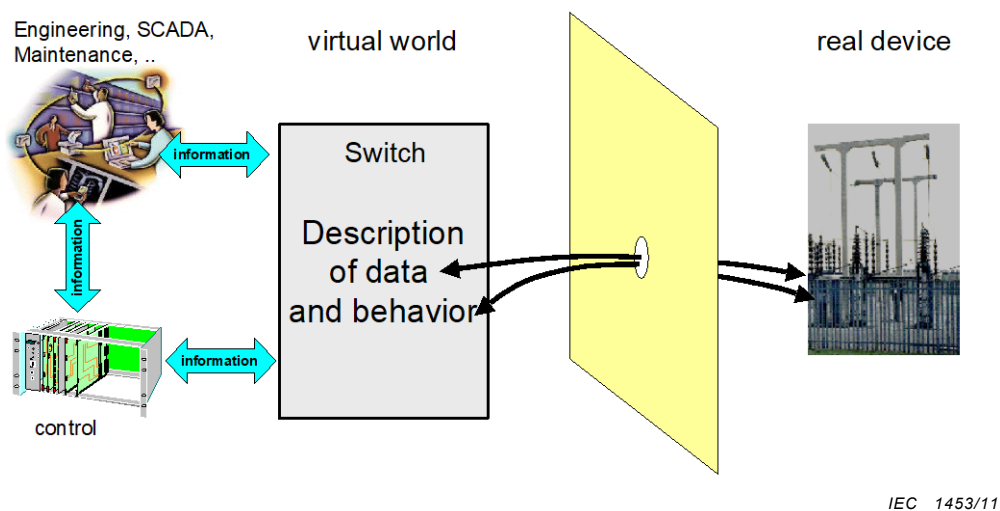
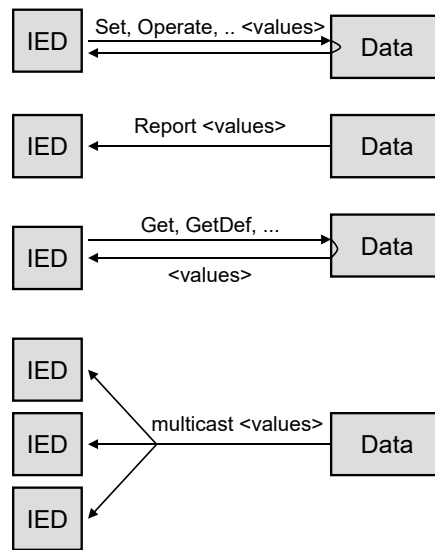


Figure 52 – Virtualisation and usage

9.4 Basic information exchange mechanisms

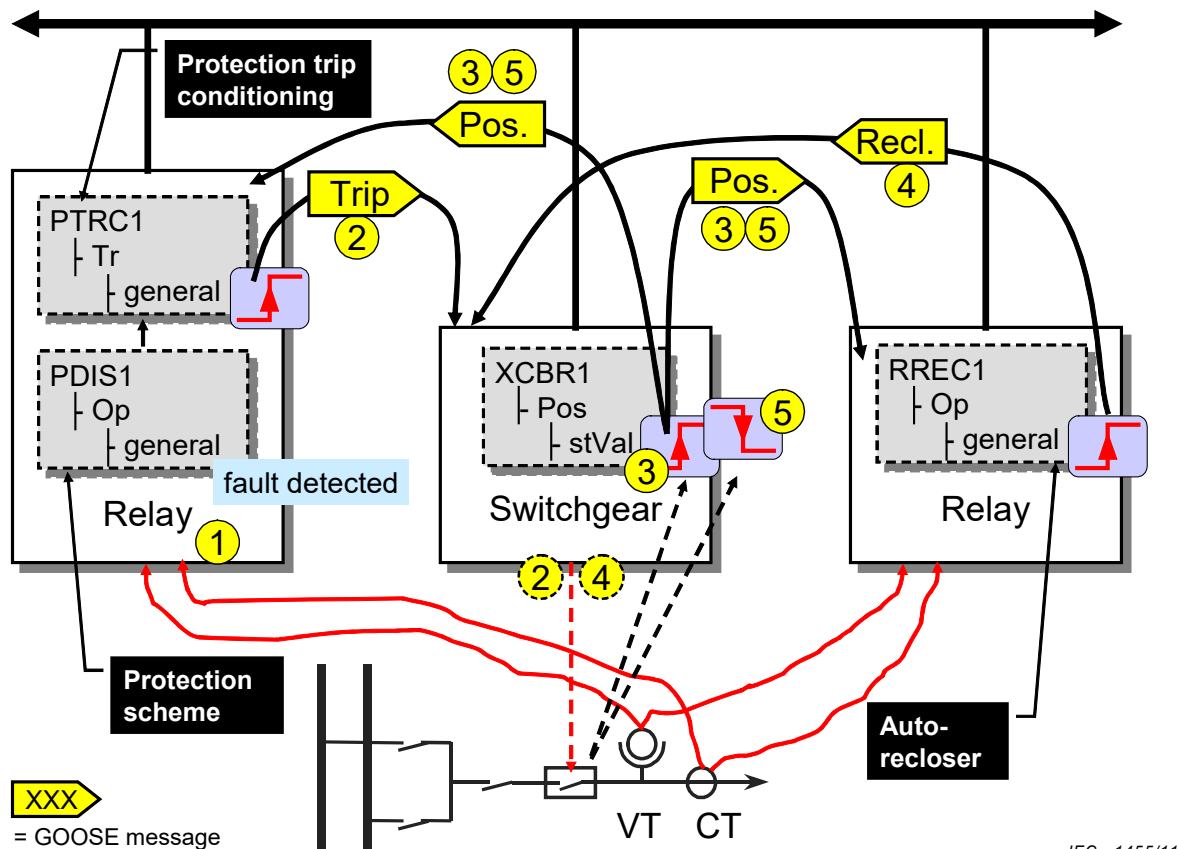
The ACSI model basically provides the methods of exchanging information between devices as shown in Figure 53.



IEC 1454/11

Figure 53 – Information flow and modelling

The use of the generic substation event model (GSE) is quite important because this model supports the implementation of real-time applications. Figure 54 shows an example of an application of the GSE model.



IEC 1455/11

Figure 54 – Application of the GSE model

Five logical nodes are involved in the example. The sequence of actions and GOOSE messages is as follows.

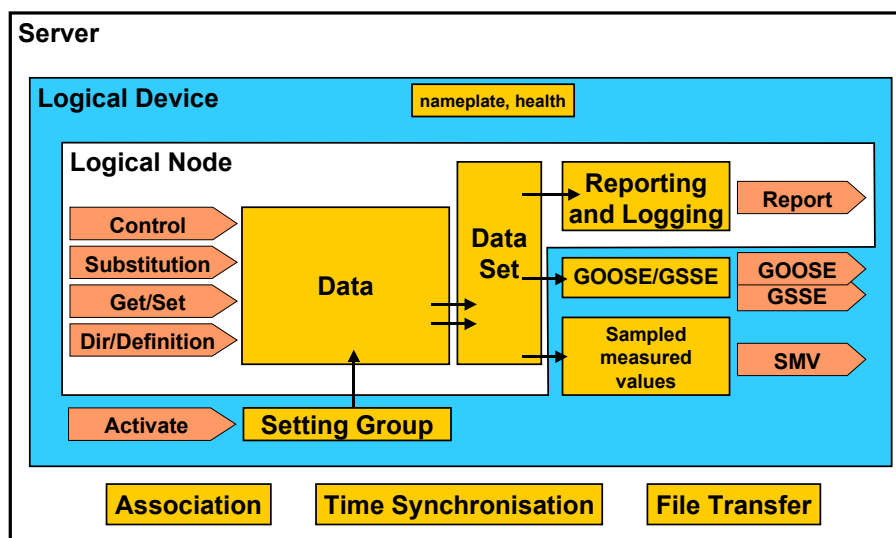
- a) The logical node “protection scheme” (PDIS1) detects a fault, this results in a decision to issue a trip.
- b) The logical node “protection trip conditioning” (PTRC1) issues a trip message (applying a GOOSE message), the “circuit breaker” (XCBR1) has been configured to receive the trip message. After additional processing, the switchgear opens the circuit breaker.
- c) The status information of the “circuit breaker” (XCBR1.Pos.stVal) changes from ON to OFF. This new state is immediately sent by a GOOSE message with the indication: <new position of switch = open>. In addition, the reporting model may report the change.
- d) The logical node “autoreclosing” (RREC1) receives a GOOSE message from the XCBR1 with the value <open>. According to the configured behaviour, the RREC decides to re-close the circuit breaker and sends a GOOSE message with the value <reclose>.
- e) The “circuit breaker” (XCBR1) receives the GOOSE message with the value <reclose>. After additional processing, the switchgear closes the circuit breaker. The XCBR1 issues sending another GOOSE message < new position of switch = close>.

The sequence is an example only. The IEC 61850 series provides the basic mechanisms for exchanging GOOSE messages under real-time conditions. Applications of GOOSE messaging may be as simple as described in the example. But it may be used in more sophisticated schemes. All these schemes are outside the scope of the IEC 61850 series.

9.5 The client-server building blocks

9.5.1 Server

Additional common building blocks provided by the communication system are depicted in Figure 55. The association model provides mechanisms for establishing and maintaining connections between devices and to implement access control mechanisms. Time synchronisation provides the accurate time for time tagging (ms range) in applications such as reporting and logging or for applications such as synchronised sampling (μ s range).

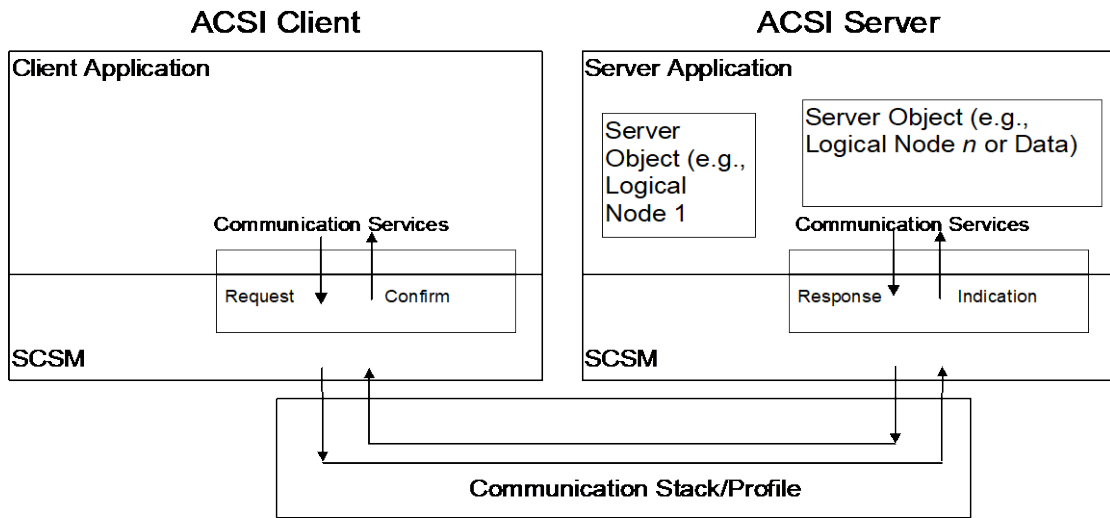


IEC 1456/11

Figure 55 – Server building blocks

The server contains everything that is defined to be visible and accessible from the communication network. A physical device may host one or more server.

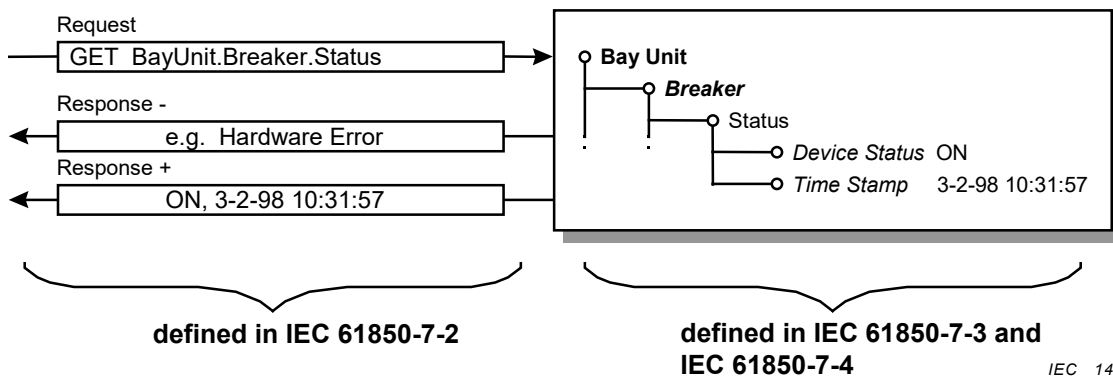
Client-server Figure 56 illustrates the client/server role. Clients issue service requests and receive confirmations of the service that has been processed in the server. A client may also receive report indications from a server. All service requests and responses are communicated by the protocol stack that is being used by a specific communication service mapping.



IEC 1457/11

Figure 56 – Interaction between application process and application layer (client/server)

Figure 57 shows an example of a get service that enables a client to retrieve the values of the data inside the server.

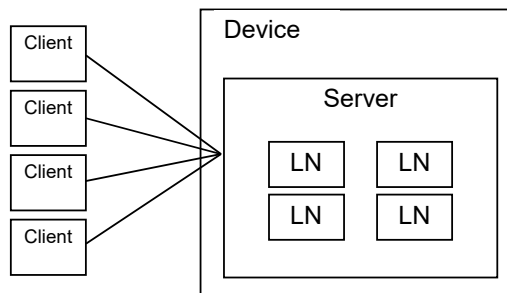


IEC 1458/11

Figure 57 – Example for a service

9.5.2 Client-server roles

According to Figure 58 one server “serves” various logical nodes and clients.



IEC 1459/11

Figure 58 – Client/server and logical nodes

The standard defines just the server role: the logical nodes, data, control, etc. located in the server, and the service request exchanged. The client role is complementary.

NOTE Clients and their internal structure and functions are not defined in this standard.

As shown in Figure 59, the devices may implement the client and the server role.

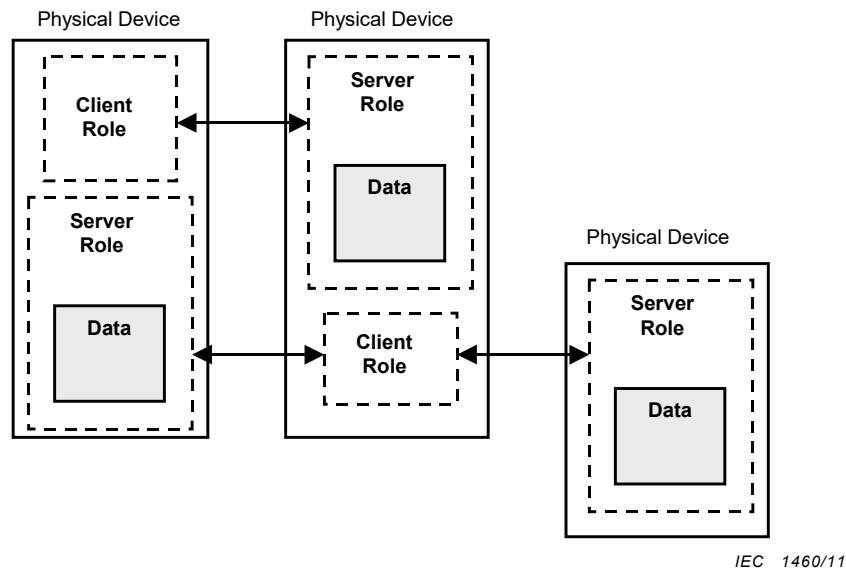


Figure 59 – Client and server roles

9.6 Logical nodes communicate with logical nodes

Logical nodes communicate with other logical nodes by means of PICOMs as described in IEC 61850-5. The logical nodes in that sense comprise the data and control as well as the client/server and publisher/subscriber roles (see Figure 59). The client and server are communication-specific entities. From an application viewpoint, they are not required. Therefore, the logical nodes (and only the logical nodes) can be understood as being in communication with one another. The logical node view and the communication view are two different views of the very same real subject.

As previously mentioned, the standard defines the server roles, i.e. the data model and the services used to exchange information. It means that only the server side of the communication view is standardized. Nevertheless, in order to describe the dataflow between logical nodes, the logical node class definition allows for the description of LN input references on the client/subscriber side as shown in Figure 60.

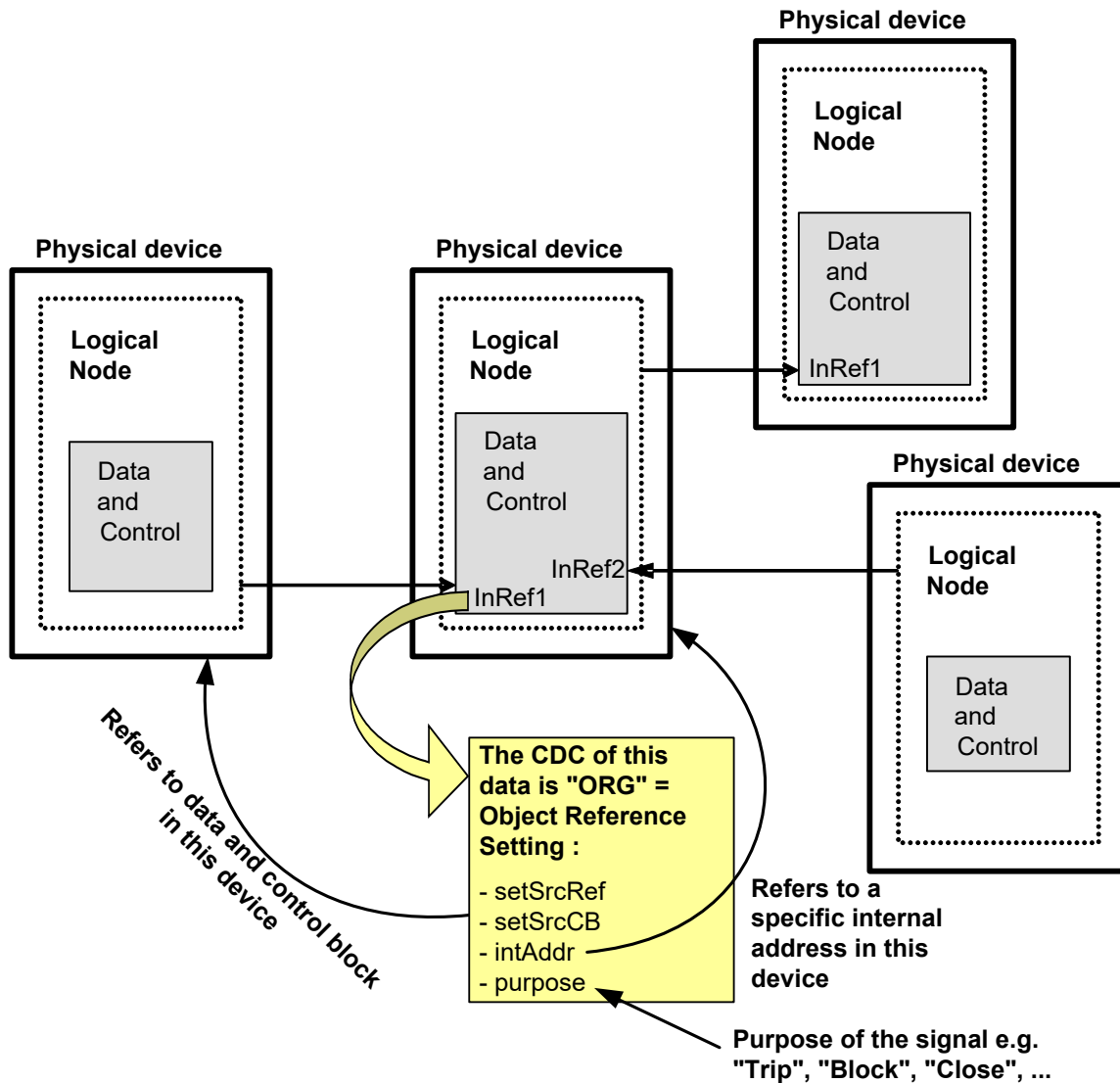


Figure 60 – Logical nodes communicate with logical nodes

Generic LN input references are used to describe incoming signals from an external source, i.e. reference to data coming from an LN hosted by the same IED or by another IED and its binding to an IED specific internal address. It is also possible to describe the intended purpose of the incoming signal. LN input references are defined by the instantiation of data named InRef and the use of the common data class ORG, “object reference setting group”.

NOTE Being a setting, the value of the CDC ORG attributes may be changed online.

Note that the SCL configuration language (see IEC 61850-6) allows the configuration of the data flow between LNs using elements “ExtRef” (see Annex H).

9.7 Interfaces inside and between devices

Real substation systems have many interfaces – interfaces for different purposes (see Figure 61). IEC 61850-7-x and IEC 61850-8-x series as well as IEC 61850-9-x series define interfaces between devices (between two devices in a client/server relationship and between many devices in a peer-to-peer relationship). IEC 61850-7-x series define abstract interfaces, while IEC 61850-8-x and IEC 61850-9-x series define concrete interfaces.

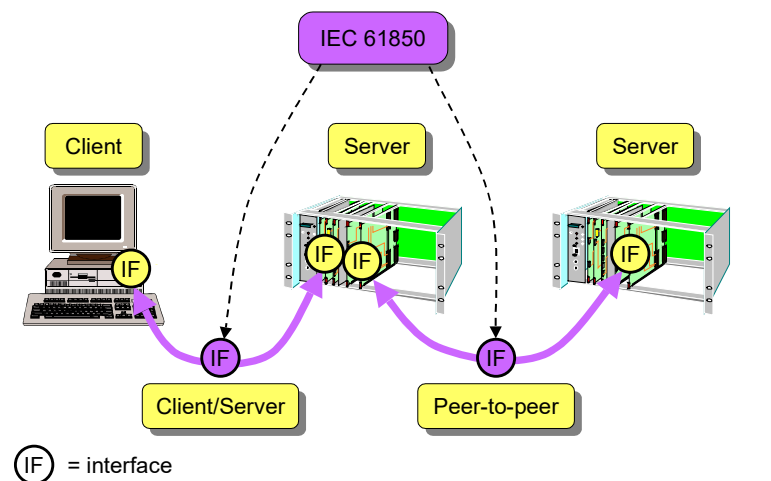


Figure 61 – Interfaces inside and between devices

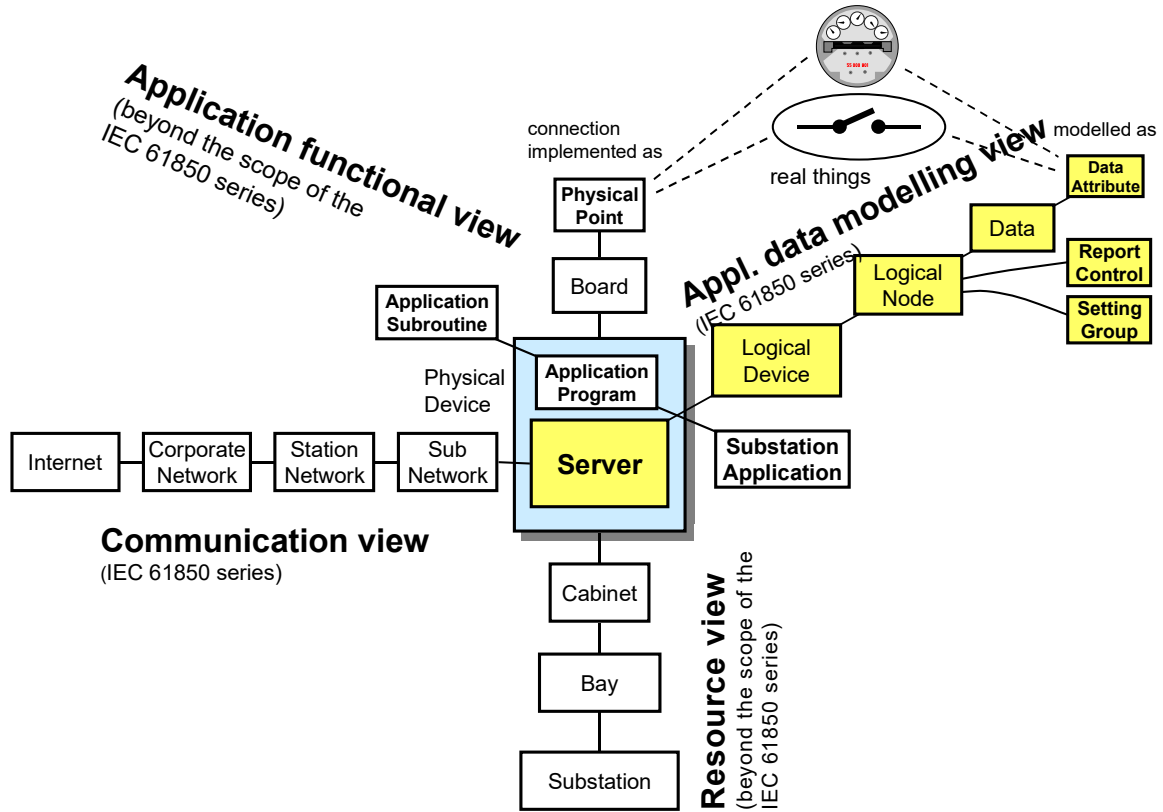
Any other interfaces (especially APIs within client or server devices) are outside the scope of this standard. On the other hand, the information model and services defined have an impact on the software and the concrete interfaces in real devices.

10 Where physical devices, application models and communication meet

Physical devices are placed in the centre of a hierarchy of components as shown in Figure 62. All views “meet” in the server. Each view has a relation to the other views inside the physical device. The various views are shown here to demonstrate that in addition to the IEC 61850 series (which describes just one view of a real automation system), many other aspects have to be taken into account when real devices are implemented.

The server is the key component. It is important to differentiate the following aspects:

- a) the server represents the application data modelling view (the IEC 61850 series) to the outside network,
- b) the server represents all aspects of the communication network and the process I/Os to the application of the physical device,
- c) a SCSM maps the IEC 61850 series view to the communication network visible objects,
- d) the server, the SCSM and the application functional view are mapped to the resource of a physical device.



IEC 1463/11

Figure 62 – Component hierarchy of different views (excerpt)

For real devices, all aspects (applications, APIs, views, mappings, relations) have to be implemented. Devices conforming to the IEC 61850 series make the IEC 61850 series view visible to any other device connected to the network for interoperability with applications running in these devices. Anything that is not modelled as a service, logical device, logical node, data, data attribute, setting group, report control, etc. is not visible to the network.

NOTE 1 The standard covers those definitions (information models and service models) that are defined to be compatible. Real devices usually require also vendor and user specific definitions that go beyond the standard. These specific definitions (outside of the scope of this standard) may be implemented as well.

NOTE 2 The engineering and configuration of real devices and systems deal with (1) the compatible definitions (mainly information models) of this standard which are covered by the SCL and (2) the application, vendor, and user specific definitions which require special attention (the extensions of the information model may be partly specified in an SCL extension).

Additional views such as the configuration view are outside the scope of this part of IEC 61850. The network management view and the system management view are not covered by this standard. A lot of information required for device management is modelled in IEC 61850-7-4 as data classes in logical node zero (LLN0). For details on the configuration view, refer to IEC 61850-6.

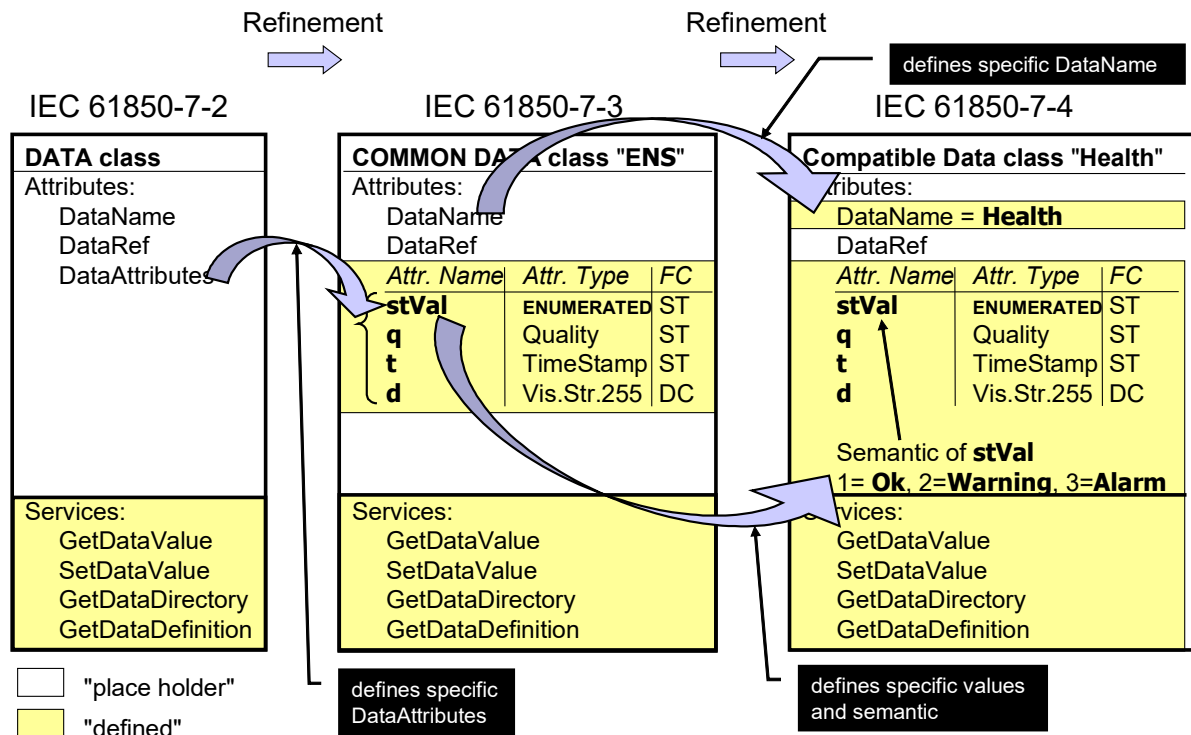
11 Relationships between IEC 61850-7-2, IEC 61850-7-3 and IEC 61850-7-4

11.1 Refinements of class definitions

One major building block is the DATA class defined in IEC 61850-7-2. The DATA class is used in the definition of almost all information that is defined in logical nodes. The DATA class as defined in IEC 61850-7-2 is on the left hand side of Figure 63. The DATA class defines three data attributes and four services. The services are defined in IEC 61850-7-2. The content of the data attributes are not specified in IEC 61850-7-2. The DATA class therefore is very generic. It must become more specific if it is to be used in an application domain. This could require definition of all DATA needed to model substation specific functions inside the

logical nodes. It is common practice to analyse an application domain to find common properties and terms applicable to many data classes. These common definitions are provided by the common data classes (CDC) specified in IEC 61850-7-3.

Common data classes are built on DATA classes. In the middle of the figure, the example common data class "ENS" (enumerated status) is shown as a refinement of the DATA class. The "ENS" refines the DataAttributes that are left empty in IEC 61850-7-2. Four attributes are defined: "stVal" (Status value), "q" (Quality), "t" (Timestamp), and "d" (Description). This common definition is used in many data definitions throughout IEC 61850-7-4.



IEC 1464/11

Figure 63 – Refinement of the DATA class

The DATA so far does not tell much about its use or the semantics of the data attributes derived from "ENS". The class on the right-hand side of Figure 63 defines exactly this "use". The "Health" class defines the name "Health". That name will be used in all instances derived from this class. In addition, the status value "stVal" is an enumerated value defined as having three values: "Ok" (=1), "Warning" (=2), and "Alarm" (=3).

The standardised names and the semantic definitions associated with the names contribute essentially to the requested interoperability.

The final definition as to what the names "OK", "Warning", and "Alarm" really mean, depends on the context in which this class is used. In a circuit breaker, it may have a slightly different meaning than in a measurement unit.

11.2 Example 1 – Logical node and data object class

Table 6 shows an example of a list of data object classes for a circuit breaker. The name of the circuit breaker class is "XCBR". The data object classes that make up the circuit breaker are grouped into four categories (descriptions, status information, controls, and settings). Each category comprises some data object classes, for example, "Mode" and "Switch position". These data object classes are referenced by their data object name: "Mod" and

"Pos". To be more precise, each data object class also has a common data class, defining the details, i.e. the attributes of the data object class. The last column specifies the presence conditions that can be applicable to the data object class, i.e. whether this data object class is mandatory (M), optional (O), in some cases forbidden (F) or can be instantiated many times (multi). This column specifies two contexts for the presence conditions: nds for the common context of Non Derived Statistical case and ds for the specific context where the LN instance is derived from another one (Derived Statistical). Refer to IEC 61850-7-4 for a detailed explanation. It follows that for each data object two presence conditions are expressed, one object in case of nds context and a second in case of ds context. For example, the data object Pos is mandatory (M) in the nds context and forbidden in the ds context (F).

Table 6 – Logical node circuit breaker

Logical Node: Circuit breaker		Name: XCBR	
Data object Class	Data object Name	Common Data Class (CDC)	PresCond nds/ds
Descriptions			
Name plate of the logical node.	NamPlt	LPL – Logical node name plate	MONamPlt/ MONamPlt
Name plate of external (electrical, mechanical or communication) equipment to which the logical node is associated.	EEName	DPL – Device name plate	O/F
Controllable Data			
(controllable) Operating mode of the domain logical node that may be changed by operator. Processing of the quality status ('q') of the received data is the prerequisite for correct interpretation of the operating mode.	Mod	ENC – Controllable integer status	O/O
(controllable) Circuit breaker/switch position.	Pos	DPC – Controllable Double Point	M/F
(controllable) If true, 'open' action has been blocked; can be set from another logical node. Operating capability ('XCBR.CBOPCap' and 'XSWI.SwOPCap' for circuit breaker and switch, respectively) does not reflect the blocked opening.	BkOPn	SPC – Controllable single point	M/F
(controllable) If true, 'close' action has been blocked; can be set from another logical node. Operating capability ('XCBR.CBOPCap' and 'XSWI.SwOPCap' for circuit breaker and switch, respectively) does not reflect the blocked closing.	BkClS	SPC – Controllable single point	M/F
(controllable) If true, the charger motor, used to prevent overload of the power supply after a busbar trip, has been enabled.	ChMotEna	SPC – Controllable single point	O/F
(controllable) If true, control authority is at station level and control from remote is disabled; otherwise control from remote is allowed.	LocSta	SPC – Controllable single point	O/F
Status information			
Read-only value, describing the behaviour of a domain logical node. It depends on the current operating mode of the logical node ('DomainLN.Mod'), and the current operating mode of the logical device that contains it ('LLNO.Mod'). Processing of the quality status ('q') of the received data is the prerequisite for correct interpretation of 'DomainLN.Beh'.	Beh	ENS – Integer status	M/M
Reflects the state of the logical node related hardware and software. More detailed information related to the source of the problem may be provided by specific attribute of the logical node.	Health	ENS – Integer status	O/O

Logical Node: Circuit breaker		Name: XCBR	
Data object Class	Data object Name	Common Data Class (CDC)	PresConds/ds
If true, the control behaviour is allowed at this level.	Loc	SPS – Single point status	M/F
State of external (electrical, mechanical or communication) equipment to which the logical node is associated.	EEHealth	ENS – Integer status	O/F
Count of operations; not resettable from remote, but may be reset from local.	OpCnt	INS – Integer status	M/O
If true, the operation has been switched (from remote) to local. This changeover is always done locally with a physical key or toggle switch, which may have a set of contacts from which the position can be read.	LockKey	SPS – Single point status	O/F
Physical capability of the circuit breaker to operate, which reflects the available energy in the drive.	CBOpCap	ENS – Integer status	O/F
Point on wave switching capability	POWCap	ENS – Integer status	O/F
Operation capability available when the switch mechanism is fully charged, as the maximum number of close-open cycles to be performed. For example when one close-open cycle can be performed, 'MaxOpCap'=1.	MaxOpCap	ENS – Integer status	O/O
Setting information			
Closing time of the breaker. The time is used to compensate for the breaker closing time, i.e., the closing command will be given a defined time before phase coincidence. This is a property of the breaker that is subject to ageing.	CBTmms	ING – Integer setting	O/F

Since many dt object classes use the same details (ATTRIBUTES), these details are therefore collected for re-use in common data classes (common to many dat object classes). The common data classes are defined in IEC 61850-7-3. As an example, the “controllable double point” (DPC) common data class for "Pos" is shown in Table 7.

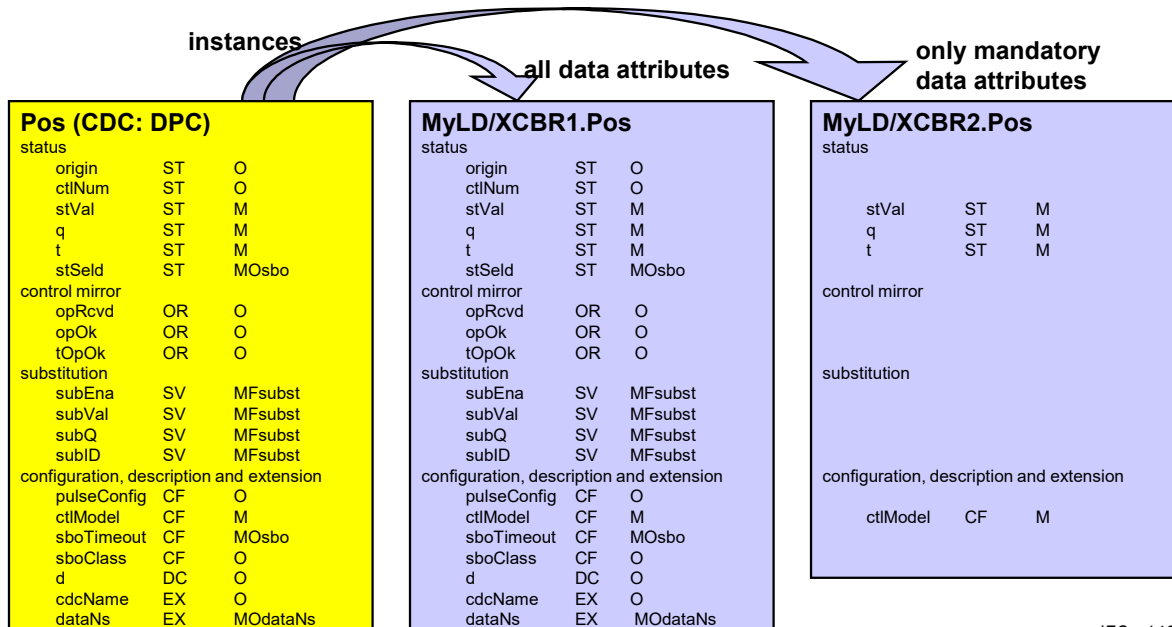
Table 7 – Controllable double point (DPC)

cdcId = DPC, UML class name = DPC					
Attribute name	Attribute type	FC	TrgOp	(Value/Value range) Description	PresCond
DataAttribute for status					
origin	Originator	ST		See 'SPC.origin'.	O
ctlNum	INT8U	ST		See 'SPC.ctlNum'.	O
stVal	DpStatusKind	ST	dchg	Status value of the controllable data object.	M
q	Quality	ST	qchg	Quality of the value in 'stVal'.	M
t	Timestamp	ST		Timestamp of the last change of the value in any of 'stVal' or 'q'.	M
stSeld	BOOLEAN	ST	dchg	See 'SPC.stSeld'.	MOsbo
DataAttribute for control mirror					
opRcvd	BOOLEAN	OR	dchg	inherited from: ControlTestingCDC	O
opOk	BOOLEAN	OR	dchg	inherited from: ControlTestingCDC	O
tOpOk	Timestamp	OR		inherited from: ControlTestingCDC	O
DataAttribute for substitution and blocked					
subEna	BOOLEAN	SV		inherited from: SubstitutionCDC	MFsubst
subVal	DpStatusKind	SV		Value used to substitute 'stVal'.	MFsubst
subQ	Quality	SV		inherited from: SubstitutionCDC	MFsubst
subID	VisString64	SV		inherited from: SubstitutionCDC	MFsubst
blkEna	BOOLEAN	BL		inherited from: SubstitutionCDC	O
DataAttribute for configuration, description and extension					
pulseConfig	PulseConfig	CF	dchg	See 'SPC.pulseConfig'.	O
ctlModel	CtlModelKind	CF	dchg	See 'SPC.ctlModel'.	M
sboTimeout	INT32U	CF	dchg	See 'SPC.sboTimeout'.	MOsbo
sboClass	SboClassKind	CF	dchg	(default=operate-once) See 'SPC.sboClass'.	O
operTimeout	INT32U	CF	dchg	See 'SPC.operTimeout'.	MOenhanced
d	VisString255	DC		inherited from: BasePrimitiveCDC	O
dU	Unicode255	DC		inherited from: BasePrimitiveCDC	O
cdcName	VisString255	EX		inherited from: BasePrimitiveCDC	O
dataNs	VisString255	EX		inherited from: BasePrimitiveCDC	MOdataNs
Parameters for control services					
ctlVal	BOOLEAN			Service parameter that determines the control activity ('FALSE' for off, 'true' for on).	

The "DPC" common data class is composed of a list of about 20 data attributes and optionally parameters for control services. Each attribute has a name, type, functional constraint, trigger option, value/value range, and an indication of whether the attribute is mandatory or optional.

At least all the mandatory attributes of all mandatory Data object classes of the "XCBR" in Table 6 make up the attributes of the "XCBR". Optional Data object classes (for example, Point on wave switching capability – POWCap) and optional data attributes (for example, origin – Originator) shall be implemented if relevant by an application or requested by a specification.

All (possible) data attributes of the data object class Pos derived from the common data class DPC are shown on the left hand side of Figure 64. An instance containing all data attributes is depicted in the middle. The Data object classes Pos is contained in the logical device MyLD and in the logical node XCBR1. The second instance has just the four mandatory data attributes.



IEC 1465/11

Figure 64 – Instances of a Data object class (conceptual)

During system design, the designer shall decide which data attributes are required to meet the required functionality of a logical node.

The conditions in the last column of the Data object class Pos are as follows (excerpt of IEC 61850-7-2 shown):

Abbreviation	Condition
M	Attribute is mandatory
O	Attribute is optional
MFsbo	Element is mandatory if declared control model supports 'sbo-with-normal-security' or 'sbo-with-enhanced-security', otherwise forbidden.
MFsubst	Element is mandatory if substitution is supported (for substitution, see IEC 61850-7-2), otherwise forbidden.
MFenhanced	Element is mandatory if declared control model supports 'direct-with-enhanced-security' or 'sbo-with-enhanced-security', otherwise forbidden.
MOdataNs	Element is mandatory if the namespace of its data object deviates from the namespace of its logical node, otherwise optional. See IEC 61850-7-1 for use of namespace.
...	...

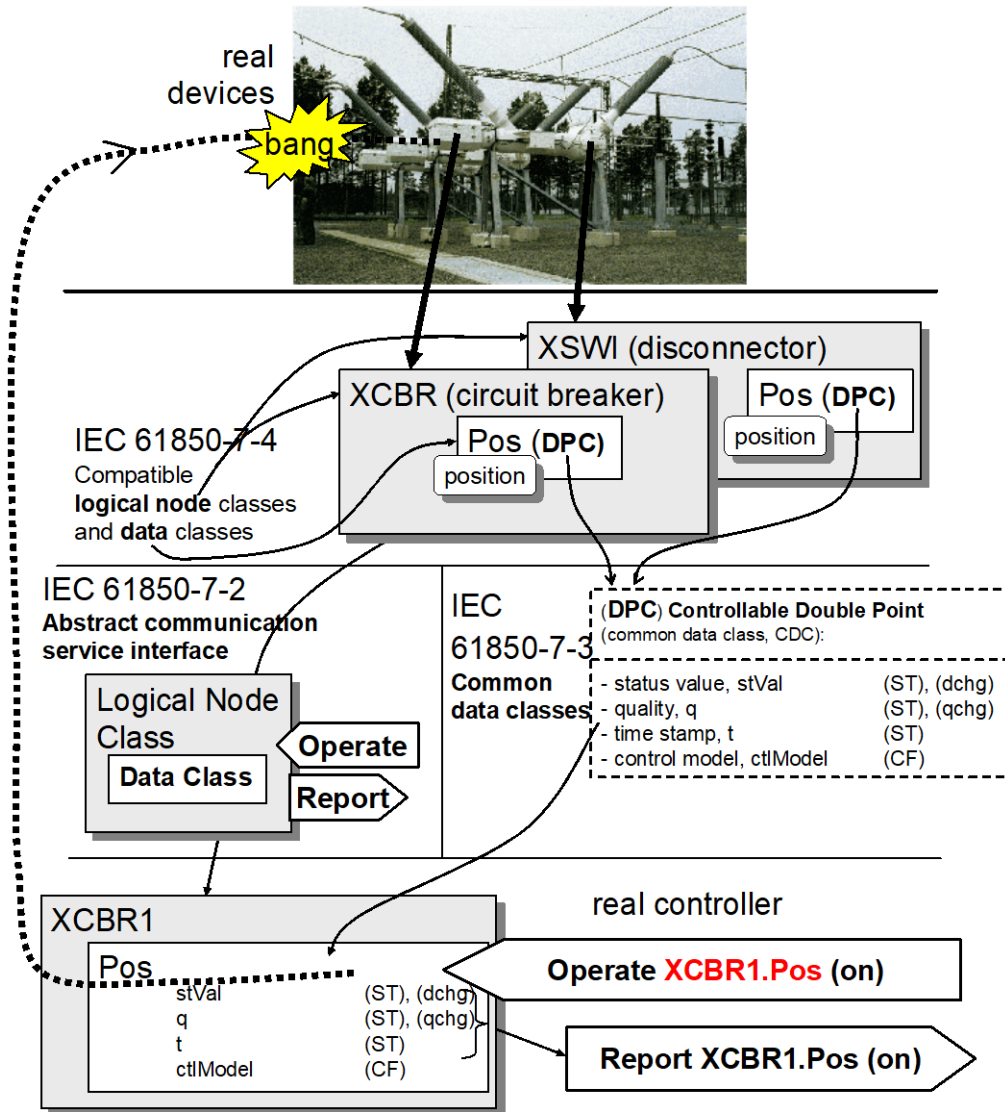
All Data object classes of the circuit breaker class together contain (i.e. when the common data classes are expanded) a total of more than 100 simple data attributes (all mandatory and optional data attributes counted).

11.3 Example 2 – Relationship of IEC 61850-7-2, IEC 61850-7-3, and IEC 61850-7-4

IEC 61850-7-4 specifies the application-specific semantic for logical node classes and the data object classes that belong to logical node classes. The data object classes represent structured information, for example, status, quality, or timestamp. A set of common simple

and complex structures applicable in most applications are defined in IEC 61850-7-3 (common data classes).

Figure 65 depicts an example of the relation between the three parts. On the level of IEC 61850-7-4, two logical node classes "XCBR" and "XSWI" are exposed. Each logical node has a data object class representing the "controllable double point position" (common data class: "DPC"). IEC 61850-7-3 defines a list of some 20 common data object classes that can be used to describe the common functionality of data. One logical node instance XCBR1 is shown at the bottom of Figure 65. This instance can be accessed by services.



IEC 1466/11

Figure 65 – Relation between parts of the IEC 61850 series

The common data class "DPC" comprises a re-usable list of attributes such as status value, quality or time stamp (values that can be reported: ST), and control model (value that can be configured: CF). The attributes of the common data classes have standardised data attribute names, for example, "stVal", or "q". These names are used in the communication (independent of the SCSMs) and in the substation configuration (language) according to IEC 61850-6.

The status value stVal has an additional information about when to trigger a report (dchg – trigger to send a report on value change of the status value). Reports can also be triggered by changes of the quality q attribute qchg.

An application of the logical node class and data object class (see IEC 61850-7-4), common data class (see IEC 61850-7-3), and the common logical node, data object class, and services (see IEC 61850-7-2) in a real system is shown at the bottom of Figure 65. The service (Operate "XCBR1.Pos" = on) closes the circuit breaker. The service (Report "XCBR1.Pos") informs the receiver about the current position change with time stamp and quality information. After a successful switching process, the "stVal" data attribute has the new state information.

12 Formal specification method

12.1 Notation of ACSI classes

IEC 61850-7-2 shall use the class notation as depicted in Table 8.

Table 8 – ACSI class definition

ABC class		
Attribute name	Attribute type	Value/value range/explanation
Attribute1 [1..n]	Type1	
Attribute2 [0..n]	Type2	
Services		
Service1		
Service2		

The class name in the tables shall be written in CAPITAL letters of Tahoma format. The attributes of a class shall have an attribute name and an attribute type. The multiplicity of the attributes shall be:

- [0..n] – attribute may be available zero to *n* times (for example DataSet [0..n] in the GenLogicalNodeClass);
- [1..n] – attribute may be available one to *n* times (for example DataObject [1..n] in the GenLogicalNodeClass);
- [0..1] – attribute may be available zero or one times (for example SettingGroupControlBlock [0..1] in the GenLogicalNodeClass).

NOTE The service parameter multiplicity (in the service tables) applies the same syntax.

The services defined for a class shall be contained in the last row.

In the text, all class names shall be bold letters of Tahoma format (for example **GenLogicalNodeClass**) to differentiate “normal” text and standardised (reserved) names. In addition, other key words such as attribute names etc. shall be in bold letters of Tahoma format (for example **LNRef**).

12.2 Class modelling

12.2.1 Overview

The IEC 61850-7-x series uses an approach based on object modelling to describe the object and service models. In this modelling technique, object classes, the characteristics of such classes, and services (methods) on those classes are described. The defined classes aid in the understanding of the intent of IEC 61850-7-2 service procedures and their effects. In implementing IEC 61850-7-2, these classes are mapped to specific communication systems (SCSMs). A real system maps the concepts described in the model to the real device. Hence, as viewed externally, a device that conforms to IEC 61850-7-2 and a specific SCSM shall exhibit the characteristics described in these class models.

Figure 66 depicts the class model of the IEC 61850-7-x series. The generic model on the left is defined in IEC 61850-7-2. The GenServer class, which represents the external behaviour of a device, is composed of (1 to n) GenLogicalDevice classes. GenLogicalDevice classes comprise two or more GenLogicalNode classes (2 to n).

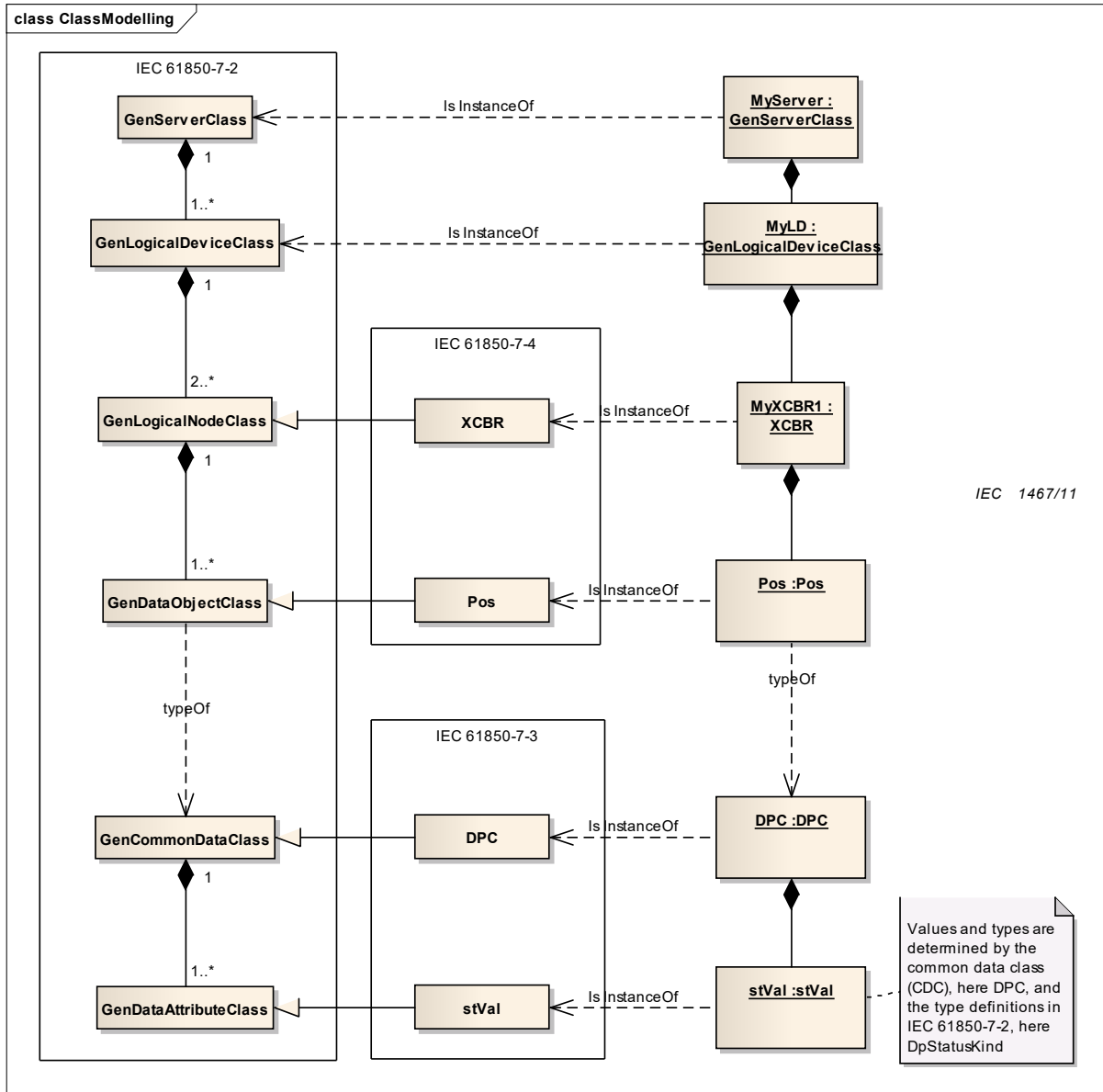


Figure 66 – Abstract data model example for IEC 61850-7-x

The logical node myXCBR1 is an instance of the class XCBR which is defined in IEC 61860-7-4. XCBR has a lot of data objects. myXCBR1 may implement just a subset of XCBR. The data object instance Pos is an instance of the data object class Pos also defined in IEC 61860-7-4. The classes XCBR and Pos are respectively derived from the GenLogicalNode class and the GenDataObject class defined in IEC 61850-7-2. The data object instance Pos is of type DPC which is a Common Data Class defined in IEC 61860-7-3. Again, here, DPC has many DataAttributes and Pos may implement just a part of the whole set. Finally, the DataAttribute instance stVal has a defined type which is determined by the common data class defined in IEC 61860-7-3 but also by the type definitions found in IEC 61860-7-2 (DpStatusKind "Double point status").

Each class is characterised by a number of attributes that describe the externally visible feature(s) of all instances of this class. Each instance of a class uses the same attribute

types, but has specific values (the instance-specific values) for the attributes. The types of these attributes are defined by IEC 61850-7-x series or may be established by IEC 61850-7-x services; hence a change in the device may be modelled by a change in one or more attribute values of an instance.

The following subclauses discuss examples of the structure of a class defined in the IEC 61850 series.

12.2.2 Common data class

The attributes of the single point status class are (according to IEC 61850-7-3) defined as depicted in Table 9.

Table 9 – Single point status common data class (SPS)

cdclid = SPS, UML class name = SPS					
Attribute name	Attribute type	FC	TrgOp	(Value/Value range) Description	PresCond
DataAttribute for status					
stVal	BOOLEAN	ST	dchg	Value of the data.	M
q	Quality	ST	qchg	Quality of the value in 'stVal'.	M
t	Timestamp	ST		Timestamp of the last change of the value in any of 'stVal' or 'q'.	M
DataAttribute for substitution and blocked					
subEna	BOOLEAN	SV		inherited from: SubstitutionCDC	MFsubst
subVal	BOOLEAN	SV		Value used to substitute 'stVal'.	MFsubst
subQ	Quality	SV		inherited from: SubstitutionCDC	MFsubst
subID	VisString64	SV		inherited from: SubstitutionCDC	MFsubst
blkEna	BOOLEAN	BL		inherited from: SubstitutionCDC	O
DataAttribute for configuration, description and extension					
d	VisString255	DC		inherited from: BasePrimitiveCDC	O
dU	Unicode255	DC		inherited from: BasePrimitiveCDC	O
cdcName	VisString255	EX		inherited from: BasePrimitiveCDC	O
dataNs	VisString255	EX		inherited from: BasePrimitiveCDC	MOdataNs

The first column represents the name of the attribute, the second specifies the attribute type. An attribute that is composed of several components is defined as depicted in the example of Table 10 (excerpt of Quality class from IEC 61850-7-2).

The components of the Quality (for example, validity or detailQual) are data attribute components. The attribute types of the data attribute components (for example, PACKED LIST or CODED ENUM) are defined in IEC 61850-7-2. An excerpt of detailQual from IEC 61850-7-2 is given in Table 16.

Table 10 – Quality components attribute definition

datId = Quality			
Attribute name	Attribute type	(Value/Value range) Description	PresCond
validity	ValidityKind	Validity of the value, as condensed information for the client. In case this value is not 'good', some reasons may be found in the 'detailQual'.	M
detailQual	DetailQual	Describes some reasons in case 'validity' is not 'good'.	M
source	SourceKind	(default=process) Defines the source of a value. NOTE 1 Substitution may be done locally or via the communication services. In the second case, specific attributes with a FC=SV are used. NOTE 2 There are various means to clear a substitution. As an example, a substitution that was done following an invalid condition may be cleared automatically if the invalid condition is cleared. However, this is a local issue and therefore not within the scope of this standard.	M
test	BOOLEAN	(default=FALSE) If true, the value is a test value. The processing of the test quality in the client shall be as described in IEC 61850-7-4. This bit shall be completely independent from the other bits within the quality descriptor.	M
operatorBlocked	BOOLEAN	(default=FALSE) If true, further update of the value has been blocked by an operator. The value shall be the information that was acquired before blocking. If this flag is set, then the 'detailQual.oldData' shall also be set. The operator shall use the data attribute 'CDC.blkEna' to block the update of the value. NOTE Both an operator as well as an automatic function may freeze communication updating as well as input updating. In both cases, 'detailQual.oldData' will be set. If the blocking is done by an operator, then this flag is set additionally, and an operator activity is required to clear the condition. EXAMPLE An operator may freeze the update of an input, to save the old value before the auxiliary supply is switched off.	M

Table 16 – Attributes of DetailQual

datId = DetailQual			
Attribute name	Attribute type	(Value/Value range) Description	PresCond
overflow	BOOLEAN	(default=FALSE) If true, the value of the attribute to which the quality has been associated is beyond the capability of being represented properly (used for measurand information only). EXAMPLE A measured value may exceed the range that may be represented by the selected data type, for example the data type is a 16-bit unsigned integer and the value exceeds 65 535.	M
outOfRange	BOOLEAN	(default=FALSE) If true, the attribute to which the quality has been associated is beyond a predefined range of values. The server shall decide if validity shall be set to invalid or questionable (used for measurand information only). EXAMPLE A measured value may exceed a predefined range, however the selected data type can still represent the value, for example the data type is a 16-bit unsigned integer, the predefined range is 0 to 40 000, if the value is between 40 001 and 65 535 it is considered to be out of range.	M
badReference	BOOLEAN	(default=FALSE) If true, the value may not be a correct value due to a reference being out of calibration. The server shall decide if validity shall be set to invalid or questionable (used for measurand information and binary counter information only).	M
oscillatory	BOOLEAN	(default=FALSE) To prevent overloading of event driven communication channels, it is desirable to detect and suppress oscillating (fast changing) binary inputs. If a signal changes in a defined time twice in the same direction (from 0 to 1 or from 1 to 0), then it shall be defined as an oscillation and this attribute shall be set to true. If a configured number of transient changes is detected, they shall be suppressed. In this time, the 'Quality.validity' shall be set to 'questionable'. If the signal is still in the oscillating state after the defined number of changes, the value shall be left in the state it was in when this flag was set. In this case, the 'Quality.validity' shall be changed from 'questionable' to 'invalid' and kept so as long as the signal is oscillating. If the configuration is such that all transient changes should be suppressed, the 'Quality.validity' shall be set immediately to 'invalid' and this flag to true (used for status information only).	M
failure	BOOLEAN	(default=FALSE) If true, a supervision function has detected an internal or external failure.	M
oldData	BOOLEAN	(default=FALSE) If true, an update is not made during a specific time interval. The value may be an old value that may have changed in the meantime. This specific time interval may be defined by an allowed-age attribute. NOTE "Fail silent" errors, where the equipment stops sending data, will cause setting this flag to true. In this case, the last received information was correct.	M

datId = DetailQual			
Attribute name	Attribute type	(Value/Value range) Description	PresCond
inconsistent	BOOLEAN	(default=FALSE) If true, an evaluation function has detected an inconsistency.	M
inaccurate	BOOLEAN	(default=FALSE) If true, the value does not meet the stated accuracy of the source. EXAMPLE The measured value of power factor may be noisy (inaccurate) when the current is very small.	M

The FC column specifies the functional constraint, if applicable. The functional constraint indicates which services can be used to access the values of the data attributes. Table 11 shows which services are allowed for the status information. This is an excerpt from IEC 61850-7-2.

Table 11– Functional constraints (excerpt)

Functional constraints		
FC	Semantic	Description (services allowed, initial values, storage)
ST	Status information	Data attribute shall represent status information. Initial value shall be taken from the process. Modelling note: Applicable ACSI services: - GetDataValues - GetDataDefinition - GetDataDirectory - GetDataSetValues - GetAllDataValue - may be a DataSetMember of a DataSet referred to by any of: GOOSE control block, report control block, log control block, sampled value control block.

The services applicable are listed in the third column. For all data that inherit the attributes from the common data class SPS (see Table 11), the attributes with FC=**ST** can be accessed by these services.

The trigger options TrgOp specify the possible trigger conditions that may cause to send a report or to log events. The service procedures shall be as specified in Table 12.

Table 12– Trigger option

Trigger options		
TrgOp	Semantic	Description
dchg	data-change	The change of the value of the associated data attribute can trigger the generation of a report or a log entry.
qchg	quality-change	The change of the value of the quality data attribute can trigger the generation of a report or a log entry.
dupd	data-update	The update of the value of the associated data attribute can trigger the generation of a report or a log entry. The updated value may be the same as the old value. An example is freezing the value of a freezable data attribute updating the value of another data attribute, which could lead to the same value it already has. Another example would be the distance to a fault that would persist, or a calculation of a minimum value over a period and the same minimum is reached in two consecutive statistical calculation windows.
dchg, dupd	dchg or dupd	Either 'dchg' or 'dupd', to be chosen at data object definition in the scope of a logical node, or at implementation.

As depicted in Figure 67, the value of a data attribute that provides a specific TrgOp (trigger option) shall be monitored for reporting and logging if the report control block has enabled the specific trigger option (TrgOps). In the example below, the TrgOps is dchg; the TrgOp of the data attribute is dchg for the first, dupd for the second, and qchg for the last data attribute. Reports are sent on data changes only, because only dchg is enabled in the report control block. In the second example, all changes will be reported. In addition, a report will be sent on the expiration of the integrity period.

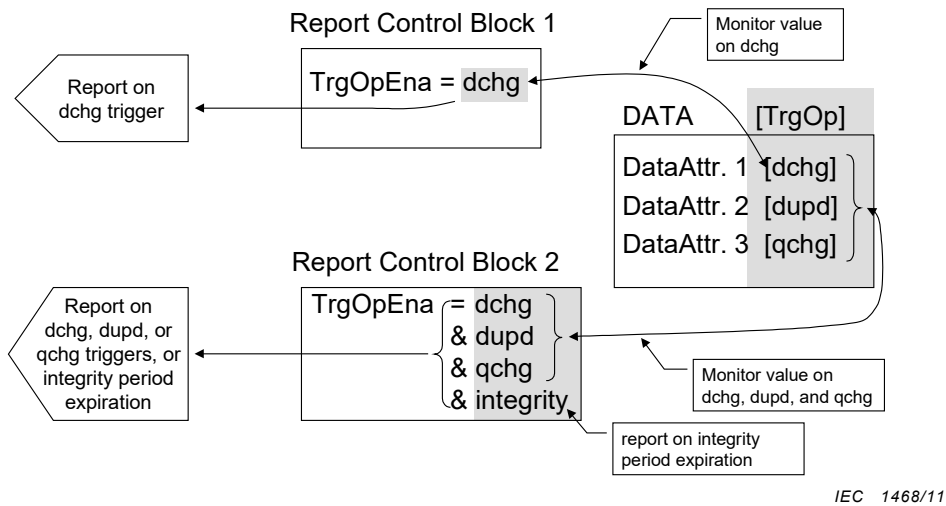


Figure 67 – Relation of TrgOp and Reporting

The column “(value/value range) Description” contains a description of the attribute and may provide some values/value ranges allowed for the attribute. The last column “PresCond” indicates the presence conditions of the attribute (optional, conditional, ...).

12.2.3 Logical node class

Table 13 shows the table of a basic logical node class defined in IEC 61850-7-2: the GenLogicalNodeClass. The logical nodes contained in IEC 61850-7-4 inherit all definitions from this basic logical node class.

Table 13 – GenLogicalNodeClass definition

GenLogicalNodeClass		
Attribute name	Attribute type	Explanation
LNNName	ObjectName	Instance name unambiguously identifying the logical node within the scope of a logical device.
LNRef	ObjectReference	Unique path-name of the logical node: LDName/LNNName.
DataObject [1..n]	GenDataObjectClass	All data objects contained in the logical node. NOTE IEC 61850-7-4 defines standardized data object classes.
DataSet [0..n]	DATA-SET	All data sets contained in the logical node.
BufferedReportControlBlock [0..n]	BRCB	All buffered report control blocks contained in the logical node.
UnbufferedReportControlBlock [0..n]	URCB	All unbuffered report control blocks contained in the logical node.
LogControlBlock [0..n]	LCB	All log control blocks contained in the logical node.
IF compatible LN class equals LLN0 of name space IEC 61850-7-4		
SettingGroupControlBlock [0..1]	SGCB	Setting group control block contained in the logical node.
Log [0..n]	LOG	All logs contained in the logical node.
GOOSEControlBlock [0..n]	GoCB	All GOOSE control blocks contained in the logical node.
MulticastSampledValueControlBlock [0..n]	MSVCB	All multicast sampled value control blocks contained in the logical node.
UnicastSampledValueControlBlock [0..n]	USVCB	All unicast sampled value control blocks contained in the logical node.
Services GetLogicalNodeDirectory GetAllDataValues		
NOTE IEC 61850-7-4 defines specialized logical node classes – the compatible logical node classes, for example, XCBR representing circuit-breakers. Domain name space owner needs to specify explicitly the presence of Log in the definition of their compatible logical node classes where the support of Log is required additionally to the LLN0 of name space IEC 61850-7-4.		

The columns of the class table are: attribute name, attribute type, and explanation.

The lines represent the attributes of the logical node.

Each logical node class has a logical node name (LNNName). IEC 61850-7-4 defines many logical node class names, for example, XCBR for the logical node “circuit breaker”.

The logical node reference (LNRef) is used to reference an instance of a logical node. An example is MyLD/XCBR1. That means there is an instance with the name XCBR1 of class XCBR that is contained in the logical device MyLD.

The logical node contains one or more data (DataObject). Data represent the function (and semantic) of the logical node. The logical nodes defined in IEC 61850-7-4 each contain a list of a few to many data.

Data sets (DataSet) contained in a logical node may reference data and data attributes included defined in the same logical node or contained in any other logical node of any logical device.

Report and log control blocks may be contained in a logical node as well. IEC 61850-7-4 does not define any common report or log control blocks nor any common data sets. It is up to the system design to define specific data sets and report and log control blocks.

The presence of the last five optional attributes is only available in the LN class LLN0.

The services that operate on the logical node are the two services listed at the end of the table (GetLogicalNodeDirectory and GetAllDataValues). All classes that are used as types (listed in the column “Attribute type”) have their own services. The class DATA has several services, for example, GetDataValues and SetDataValues.

From this point of view, the logical node comprises all services of all classes that are used to build up the logical node class.

12.3 Service parameters tables

IEC 61850-7-2 provides unconfirmed and confirmed services. The mapping of the confirmed services requires that the used application layer provides a method that serves to identify the request and the accompanying response within an association. The service parameters tables summarise the parameters that are required for the processing of a particular service:

Request		
Parameter name	Parameter type	Description
Parameter 1...		
Parameter n		
Response+		
Parameter name	Parameter type	Description
Parameter 1...		
Parameter n		
Response–		
Parameter name	Parameter type	Description
ServiceError		

NOTE The service parameters tables of the services defined in IEC 61850-7-2 do not show all parameters required in concrete interface implementations; for example the parameter “association” or “retransmission time” are not depicted in the abstract service tables. These tables are abstract – local issues and concrete protocol issues are not shown. These specific issues are not required to understand the semantic and behaviour of the service.

Usually the service parameters table provides the request and response parameters of a specific service. Each parameter and the effect this parameter has on the processing of the service is described in IEC 61850-7-2 in an abstract way. The sequences of the service primitives of confirmed services are depicted in Figure 68.

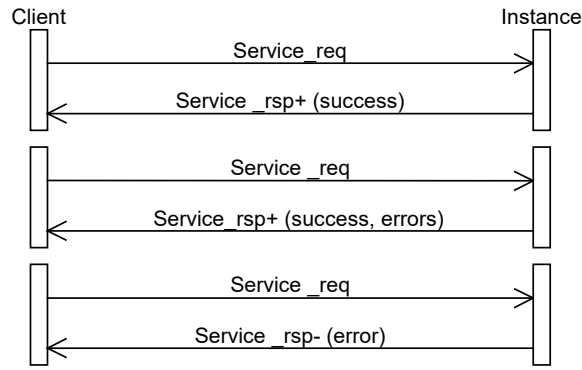
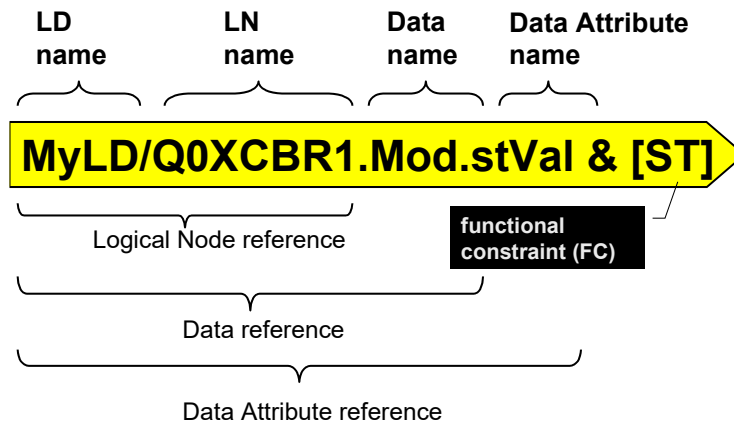


Figure 68 – Sequence diagram IEC 1469/11

12.4 Referencing instances

The standard differentiates between object names and object references. Object names (see Figure 69) identify an instance of a class at one hierarchy level (for example, "Mod" at the Data level or "Q0XCBR1" at the logical node level). "Q0" is a prefix and the "1" is a suffix to the name "XCBR". The concatenation of all the object names forms the object reference (for example, "MyLD/Q0XCBR1.Mod.stVal").



IEC 1470/11

Figure 69 – References

The data attribute reference identifies a specific data attribute of a data instance. Data reference identifies a complete data instance with all its data attributes.

The logical node name "XCBR" may be enhanced by a prefix (for example "Q0"). It shall also contain a logical node instance number (for example "1"). Those together with the logical node class name build the object name Q0XCBR1. The standardisation of prefixes and logical node instance numbers is outside the scope of this standard. All data names from IEC 61850-7-4 and data attribute names from IEC 61850-7-3 shall be used unchanged for instances. Extension rules for logical nodes, data and common data classes are specified in Clause 14 of this document.

Functional constraints (FC) play a crucial role in the definition of the information models and in the services to access the various parts of the information model. To simplify the description of service parameters, the following definitions (short cuts) have been defined to reduce the amount of parameters in a service request or response:

- functionally constrained data (FCD),
- functionally constrained data attribute (FCDA), and

- control block service parameters.

The conceptual use of FCD and FCDA is shown in Figure 70 using the MMS mapping.

NOTE MMS uses the notation “\$” and map the functional constraint FC between LNNName and DataName.

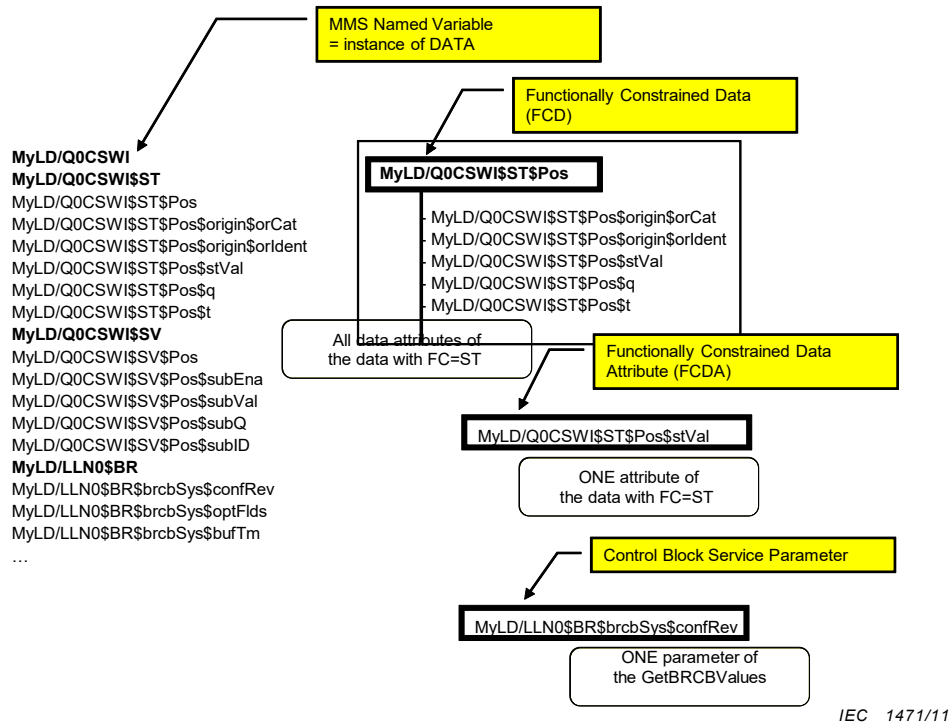


Figure 70 – Use of FCD and FCDA

As for the logical device names (MyLD in Figure 69), they are not standardized and there are no reserved names either.

The logical device names introduced in a project are to be described by the substation configuration language (SCL).

According to IEC 61860-6, SCL allows for two main options for signal naming: product-related naming and function-related naming. The former method is mandatory while the latter is optional.

For example, in case of product related naming, the name of a logical device is the name of the IED concatenated with the IED LD instance name:

S1E1Q1SB1LD2/Q0XCBR1 where:

- S1E1Q1SB1⁵ is the name of the IED,
- LD2 is the name of the IED LD instance,
- XCBR1 is the name of the LN instance,
- Q0 is the LN prefix.

⁵ SI, E1, Q1 and SB1 are names chosen according to IEC 61346-1 and IEC 61346-2 whose application is defined in IEC 61850-6.

With function related naming, the name of the LD can freely be edited. The example below shows a name based on the functional structure of the substation. In this example, the name of the logical device makes no reference to the underlying SAS.

S1E1Q1/QA1XCBR2 where:

- S1 is the name of the substation,
- E1 is the name of the voltage level,
- Q1 is the name of the bay,
- QA1 is the LN prefix corresponding to the primary equipment name⁶.

In both cases, the name of the logical device shall be unique in a subnetwork, which is easily achieved in case of product related naming.

The function-related naming convention could be used in the specification phase of the SAS. However, when implementing the specification in a real SAS, it may not be possible to apply the naming convention of the specification to the SAS. In the preceding example, Q0 will not be changed to QA1 if the former value is fixed (not configurable) in the IED S1E1Q1SB1.

Almost all services of IEC 61850-7-2 use the object references as a service parameter. These object references shall not be changed by a SCSM. They may be mapped in a SCSM to unique numbers.

Figure 71 shows examples of object names and object references. The example at the top (first four lines) can be just four class definitions (not yet instantiated) or four instances of the classes "E1QA5/XCBR.Pos.stVal", "...q", "...t", "...ctlModel". The object references do not in this case indicate if object references refer to classes or instances. The context where these references are used has to provide sufficient information to know what is meant (class or instance).

The other examples refer to instances only.

The LD name "E1QA5" and its structure are outside the scope of the IEC 61850 series. The functional constraint (FC) is not shown in the object reference. The FC information may be mapped into the object reference in a SCSM; IEC 61850-8-1 maps the FC between logical node name and the data name ("XCBR\$ST\$Pos").

⁶ SI, E1, Q1 and QA1 are names chosen according to IEC 61346-1 and IEC 61346-2.

LD	LN	Data	DAttr.	FC	
E1QA5	/XCBR	.Pos	.stVal	ST	class or instance
E1QA5	/XCBR	.Pos	.q	ST	
E1QA5	/XCBR	.Pos	.t	ST	
E1QA5	/XCBR	.Pos	.ctlModel	CF	
LD5	/YPTR2	Temp	.mVal.i .mVal.f	MX MX	instance # 2
E1QA5	/XCBR8	.Pos	.stVal	ST	instance # 8
E1QA5	/XCBR8	.Pos	.q	ST	
E1QA5	/XCBR8	.Pos	.t	ST	
E1QA5	/XCBR8	.Pos	.ctlModel	CF	
Object Name		Object Name	Object Name	Object Name	
ObjectReference					

IEC 1472/11

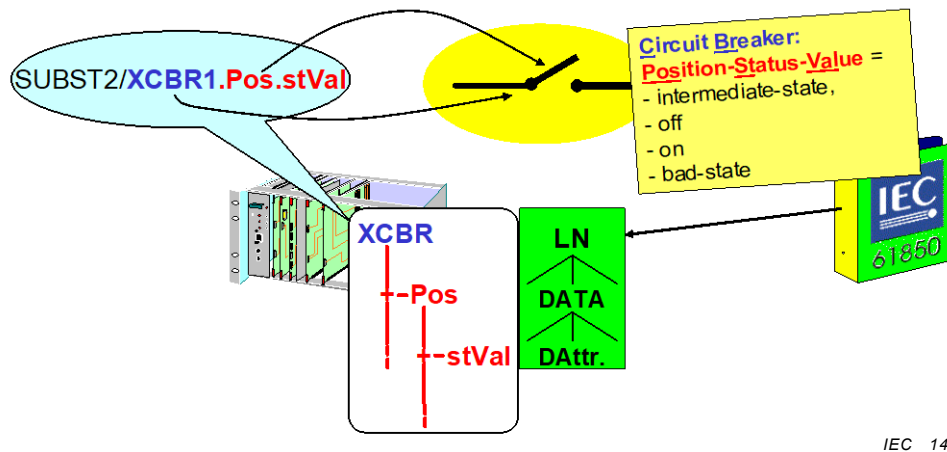
Figure 71 – Object names and object reference

13 Namespaces

13.1 General

Each class defined in IEC 61850-7-4 (for example circuit breaker LOGICAL-NODE, XCBR), IEC 61850-7-3 (for example double point status, DPS), and IEC 61850-7-2 (for example buffered report control block, BRCB) has a class name and a specific meaning associated with the class. Almost all classes are made up of other classes. The hierarchical class model of IEC 61850-7-x provides a naming hierarchy. Each name in the hierarchy has a semantic in the context where the name is used.

Figure 72 shows an example of the definition of names and their semantic in the context of a substation. The application to be modelled and defined in the IEC 61850 series may be the circuit breaker sketched in the middle at the top. The standardized circuit breaker is modelled as a LOGICAL-NODE with the specific class name XCBR. The circuit breaker is part of the LOGICAL-DEVICE with the name SUBST2. Among other attributes, the circuit breaker has the information (modelled as DATA class) that represents the position named Pos. Among other information, the position has a status (modelled as DATA-ATTRIBUTE) named stVal. The status value stVal has four values that represent the possible status of the real breaker.



IEC 1473/11

Figure 72 – Definition of names and semantics

If only the classes defined in the IEC 61850-7-x series are used to build a LOGICAL-DEVICE, then the semantic is as defined in the IEC 61850-7-x series.

For applications that need additional LOGICAL-NODEs, DATA or DATA-ATTRIBUTE rules are provided to unambiguously interpret the names, i.e., to understand the content and meaning of an instance of a class in a specific context.

Using a reference to the defined context of the data, i.e. the concept of the name space, provides a means to uniquely identify the complete semantic of an instance of a LOGICAL-DEVICE, i.e., the semantic of all its LOGICAL-NODEs, DATA, DATA-ATTRIBUTEs, and all other instances in the context of its use.

The concept of the name space allows the distinction of classes defined by different groups – as long as the namespaces have unique identifiers.

Any instance of a class of the classes defined in the IEC 61850 series and any instances of a class defined as extension to the classes of the IEC 61850 series shall provide sufficient name space information to allow the unambiguous interpretation of the semantic of the instance. The instances of classes are marked to identify the name space.

13.2 Namespaces defined in the IEC 61850-7-x series

IEC 61850-7-4 defines LN and DATA namespaces for generic application classes. IEC 61850-7-3 defines Common Data classes upon which any LN and DO namespaces rely. IEC 61850-7-2 defines a namespace for communication related (service) classes such as BUFFERED-REPORT-CONTROL-BLOCK, LOG-CONTROL-BLOCK, GenLogicalNodeClass, GenDataObjectClass, DATA-SET and for type definitions such as basic types and domain types used in the data model or as ACSI service parameters.

NOTE The mixed use of data with namespaces from other communication standards or from private definitions always implies that the conceptual approach of the data model is the same.

As depicted in Figure 73, a namespace is conceptually a class repository containing various classes. A logical device is built up by instances derived from the classes of the repository. The standard class repository that comes with the IEC 61850 series or with standards related to the IEC 61850 series is shown on the right-hand side of the figure. An example of an additional namespace is shown on the left-hand side.

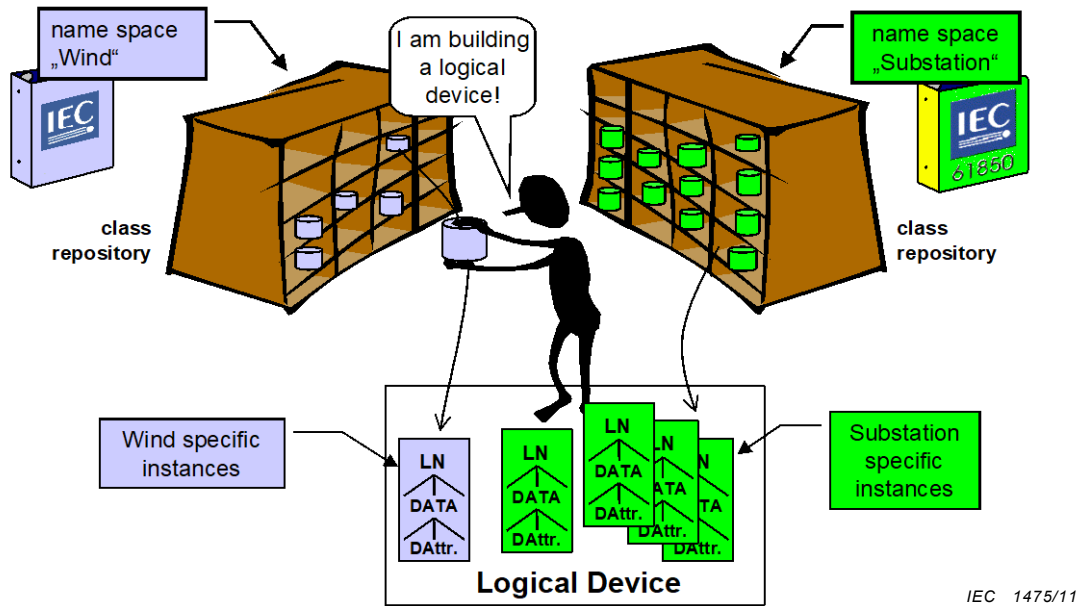


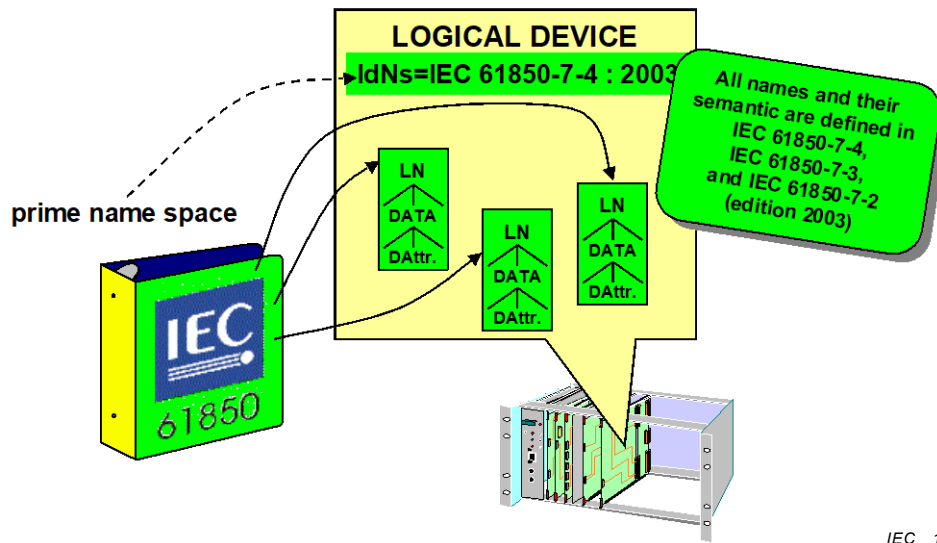
Figure 73 – Name space as class repository

The instances that are part of the logical device are coloured differently. The instances derived from the IEC 61850 series are green and the instance derived from the other standard is blue. As shown in Figure 74, the designation of the name space is represented by the attribute “logical device name space”:

IdNs = IEC 61850-7-4:2003

In this example, the name space “IEC 61850-7-4:2003” indicates that ALL instances within this logical device are derived from the 2003 editions of IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2. The logical device name space could be understood as the prime name space; even if there is just one name. The attribute IdNs (logical device name space) is an attribute contained in the name plate of the logical node zero (LLN0). The formal specification of namespaces can be found in 13.4.

IEC 61850-7-4 has normative references which apply to IEC 61850-7-3 and IEC 61850-7-2. The underlying instances of GenLogicalNodeClass, GenDataObjectClass, and GenDataAttributeClass have an implicit name space (i.e., IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2) which is derived by the normative references in IEC 61850-7-4.



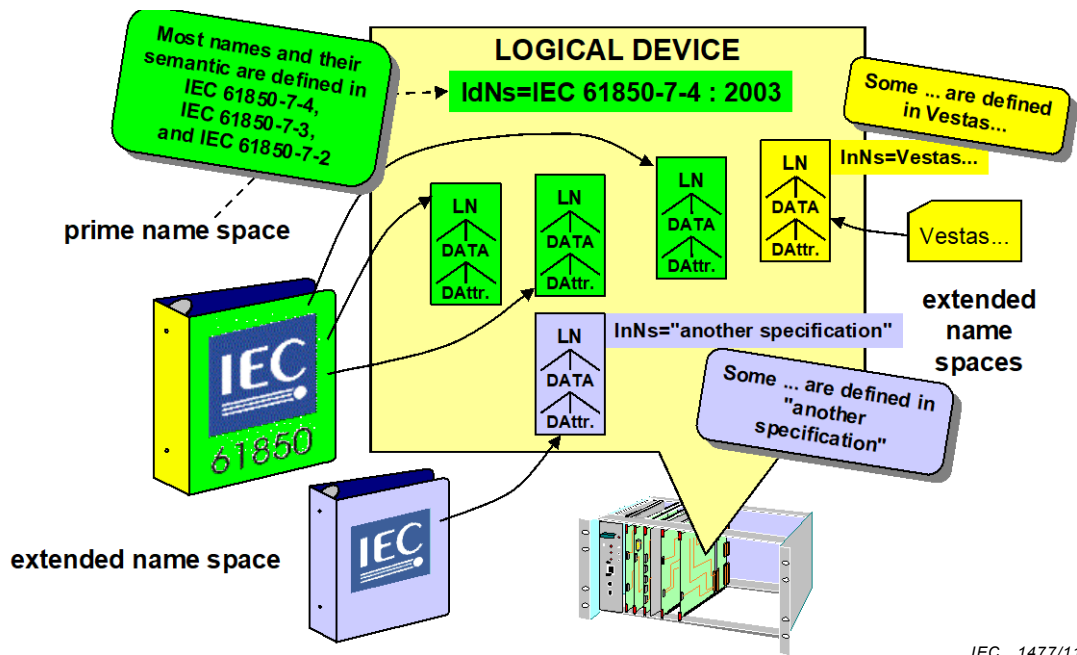
IEC 1476/11

Figure 74 – All instances derived from classes in a single name space

When additions (optional or mandatory) are defined by a next version of a standard, the standard's namespace IdNs is kept, and a version/revision identification is added, so that a human user can find the meaning. This is needed for backward compatibility between applications. This means however, that all applications use the 'may ignore' principle, i.e. ignore what they do not understand, which should normally be the case, because an automatic application will just use what it needs and knows. Problems can only come with 'generic' applications like pure communication drivers, which suddenly see new CDCs or FCs – and then have to 'ignore' them. Specific means shall be taken to deal with compatibility issues between different versions of a standard. For IEC 61850, one can find these requirements in the normative Annex K. For 'private' extensions and additional application areas, a new namespace shall always be defined.

The example instances of Figure 75 are derived from three different namespaces: IEC 61850-7-4, the other standard document, and a private name space (Vestas). Since the majority of instances are derived from IEC 61850-7-4, the logical device name space IdNs is still IEC 61850-7-4 (prime name space).

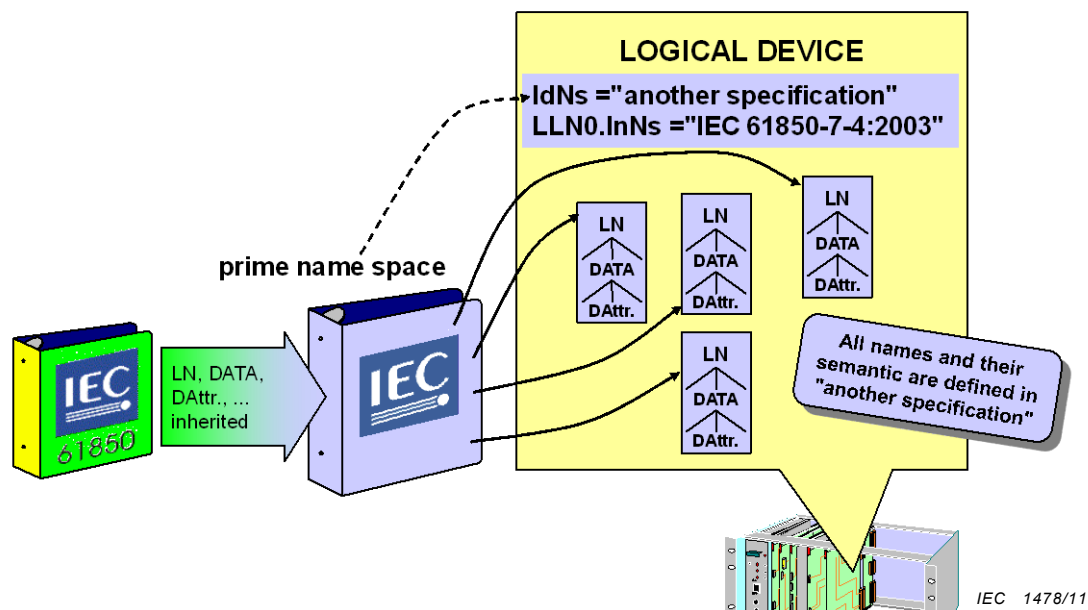
The logical node at the bottom has been derived from the other specification. This needs to be indicated with an explicit logical node name space InNs with the value, for example, "Other Specification: year of publication". The instances below that logical node are defined in the name space. The instances of data have implicitly the same name space. The third name space applied is a private name space: InNs with the value "Vestas".



IEC 1477/11

Figure 75 – Instances derived from multiple namespaces

The logical devices name space could be “IEC 61850-7-4:2003” or any other name space depending on the context in which a logical device is defined and used. The example given in Figure 76 shows how another specification could for example inherit all classes (GenLogicalNodeClass, GenDataObjectClass, and GenCommonDataClass) from IEC 61850-7-4, IEC 61850-7-3 and IEC 61850-7-2. In that case, the other specification would define a superset name space



IEC 1478/11

Figure 76 – Inherited namespaces

The IdNs has the value “other specification: year of publication”. The LLN0.InNs has the value “IEC 61850-7-4:2003” to indicate what version of the basic namespaces the logical device is based on. Since all classes are contained in these namespaces, the underlying instances have the implicitly the same name space. They do not need to have an explicit value for their

namespaces. Subclauses 13.3 and 13.4 explain in more details the relationships between namespaces owned by different entities.

This standard defines five types of namespaces:

- **Basic namespaces** contain definitions of the basic classes from which are built all other namespaces. Namely IEC 61850-7-2; IEC 61850-7-3; IEC 61850-7-4.
- **Domain namespaces** include the parts 7-4xx of the IEC 61850 series, and possibly other namespaces hosted in dedicated standards or technical specifications such as IEC 61850-80-4 and IEC 61400-25,
- **Product standard namespaces** typically contain the nameplate detailed information model, but can also be broader than specific products. IEC 62771-3 Ed 2 for switchgears or IEC 61869-9 for merging units are typical examples of product namespaces.
- **Transitional namespaces** are used in technical reports such as the 90-xx parts of the IEC 61850 series. These technical reports may extend any of the basic namespaces (IEC 61850-7-2, IEC 61850-7-3 and IEC 61850-7-4) as well as any domain namespaces. They specify extensions that are expected to be included eventually as part of the basic or domain namespaces in IEC 61850.
- **Private namespaces** are developed to provide a private data model extension (i.e. a set of non-standardized data object added to the standard one). They may be constructed to include standard common data classes from the standard namespaces or from the standard namespace extensions. They are expected to be managed by the private entity which produced them.

13.3 Namespaces dependencies

13.3.1 General

Three types of dependencies are defined:

- **Depends on:** means that a namespace version cannot exist without the classes that are defined in the namespace version it depends on. It is mainly used for basic namespaces.
- **Includes:** means that a namespace version includes some or all the classes defined in another namespace version. Applicable for domain namespaces.
- **Extends:** means that a namespace version may extend the classes defined in another namespace version.

13.3.2 Basic namespaces dependencies

NOTE The informative Annex I is a proposal for Edition 3 of this part of the IEC 61850 series to modify the dependencies between the IEC 61850 related namespaces in order to allow independent evolution of the domain application namespaces. It would replace this clause.

Within the IEC, namespaces are defined by entities (e.g. IEC TC 57) responsible for vocabularies used by specific domain applications. These entities are called owners of a name space. Entities may own several namespaces.

Figure 77 shows the dependencies between different namespaces in the context of the IEC 61850 series.

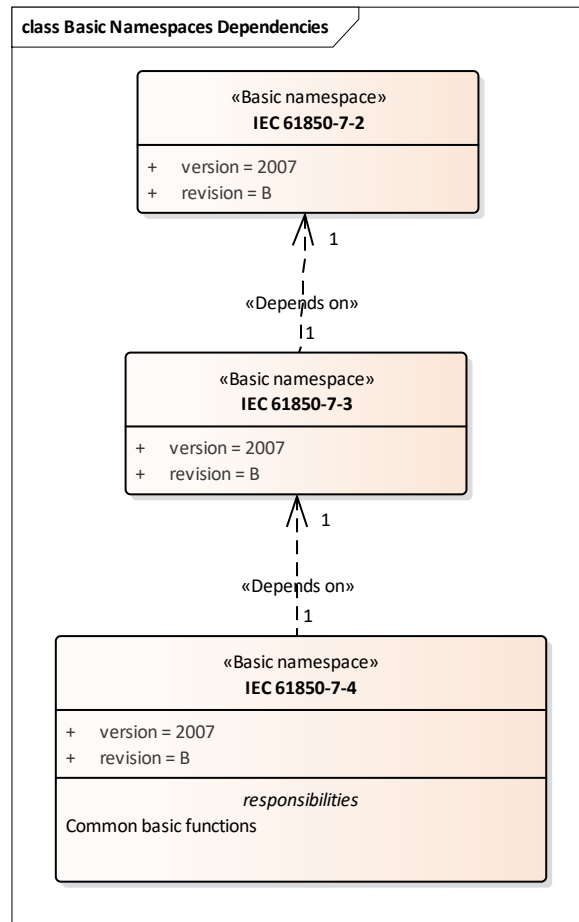


Figure 77 – Basic namespaces dependencies

The basic namespaces covered by the IEC 61850 series are:

- IEC 61850-7-2 which specifies:
 - Basic types, domain types and service types,
 - Generic classes and their associated services (logical node class, data object class, common data class),
 - Dataset, control block and control model classes,
 - Generic substation event class model (GSE),
 - Transmission of sampled value class model,
 - Time and time-synchronization model.
- IEC 61850-7-3 which specifies common data classes and enumerations.
- IEC 61850-7-4: specifies LN and enumerations common to all applications (and currently hosts the substation application domain);

The dependency “Depends on” in Figure 77 shows that IEC 61850-7-4:2007B, IEC 61850-7-3:2007B and IEC 61850-7-2:2007B shall exist as a whole.

The namespaces for domain applications covered by the IEC 61850 series are:

- IEC 61850-7-410: specifies LN and enumerations for hydroelectric power plant applications;
- IEC 61850-7-420: specifies LN and enumerations for distributed energy resource applications;

- IEC 61850-7-XXX for future applications;

13.3.3 Other namespaces dependencies

Figure 78 shows the concept of namespaces inclusion/extension.

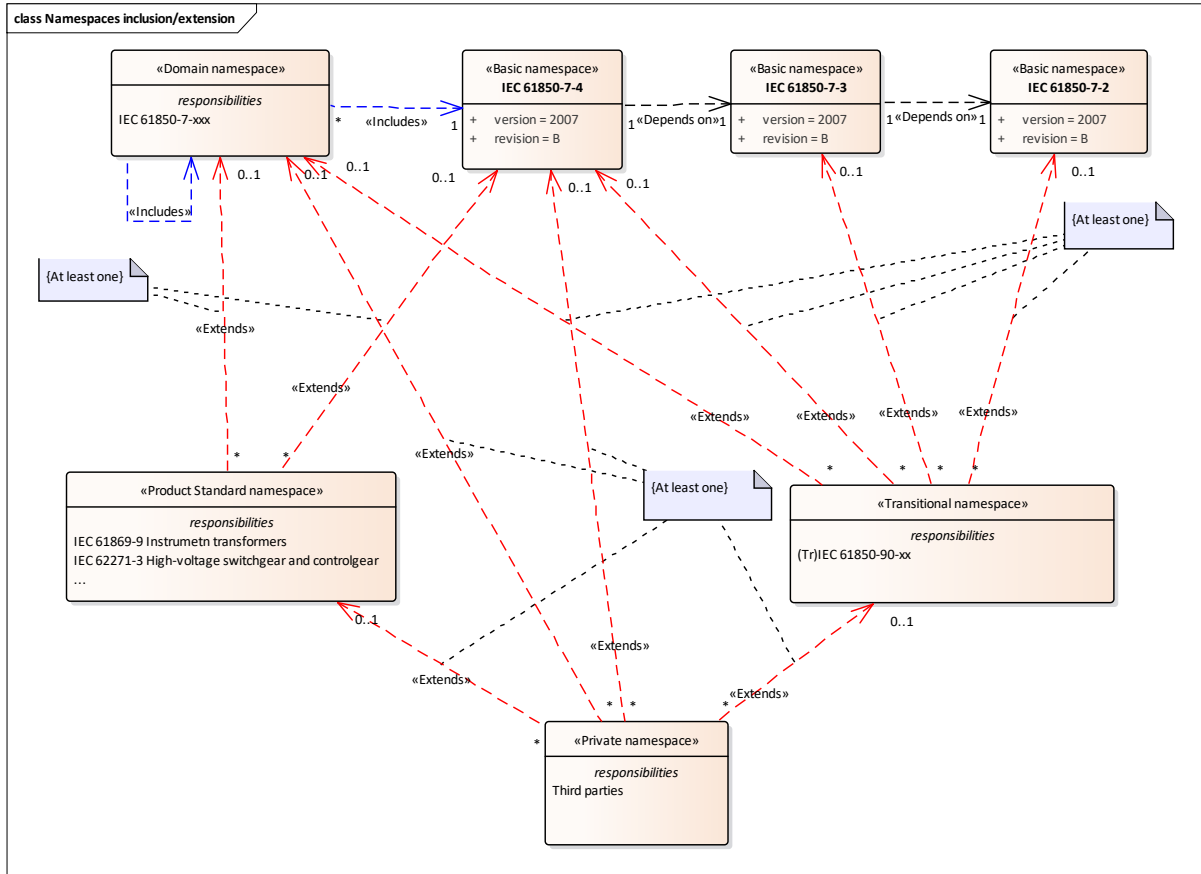


Figure 78 – Other namespaces dependencies

A domain application de facto reuses some or all classes defined in the IEC 61850-7-4 basic namespace. This is shown by the dependency “Includes”. The domain application specification shall explicitly state which edition/version and revision of IEC 61850-7-4 it includes. For example, if it is the case, IEC 61850-7-420 (Distributed Energy Resources) shall declare that it includes the IEC 61850-7-4 version 2007 revision B definitions. An example can be found in Clause J.9.

Extensions (shown by the dependency “Extends”) to the IEC 61850 series definitions may be made by different actors e.g.:

- Private extensions made by third parties.
- Extensions made through IEC 61850 Technical Reports and/or new version/revision of the IEC 61850 series definitions.
- Extensions made by TCs responsible for Product Standard owners.

As namespaces evolve with time, either for enhancements of their content or for error corrections, some rules have to be defined by owners in order to avoid compatibility issues between different versions of a namespace. For IEC 61850, these rules can be found in the normative Annex K of this document.

Also, some rules for extensibility shall be provided by owners to allow third parties to make extensions to a namespace in a way that do not jeopardize interoperability. One can refer to the specification IEC TS 61850-1-2 which is a guideline on extending IEC 61850.

Clause 14 specifies rules for class version control and extension of classes, including the use of namespaces.

13.4 Specification of namespaces

Name space names shall be structured as follows:

NamespaceID:VersionRevision

Where

- NamespaceID is the identification of the name space. NamespaceID shall refer to a technical specification (paper specification, schema (XML, database), UML description, etc.), e.g. IEC 61850-7-4,
 - In case of namespaces specified in Technical Reports (such as in the 61850-90-XXX series), the NamespaceID shall start with the prefix (Tr) to indicate that the name space shall be considered as a transitional name space, e.g. (Tr)IEC 61850-90-4:2011. In addition, the following accompanying text shall be added to the definition of the name space name:
 - a) ‘The content of this name space shall be considered as a transitional name space. This means that the content may be moved to an IS status with some changes, thus making possible backward incompatibilities with existing implementations.’
 - Version refers to the year of publication of the specification. For the domain covered by Edition 1 of IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2, the default value is 2003. For Edition 2, the value shall be 2007,
 - Revision refers to a revision indicator. This indicator is required to differentiate revisions of a specification between two versions. For a given first version, the value of this indicator shall be A and is optional, e.g. 2007 and 2007A are considered equivalent. For this amendment of Edition 2, the value shall be B and is mandatory. Subsequent revisions values shall be C, D, etc.

Name space names shall be explicitly defined within the medium describing the content (paper specification, schema (XML, database), UML description, etc.).

13.5 Attributes for references to namespaces

13.5.1 General

The following five attributes that reference namespaces are defined:

- Attribute logical device name space (ldNs) shall reference the prime technical specification used for the whole logical device.
- Attribute logical node name space (lnNs) of the LLN0 shall reference the basic name space version/revision used for the whole logical device.
- Attribute logical node name space (lnNs) shall reference the logical node name space of a single instance of GenLogicalNodeClass.
- Attribute data name space (dataNs) shall reference the data name space of a single instance of GenDataObjectClass.
- Attribute common data class name space (cdcNs). This data attribute has been deprecated.

The common data classes contain the DataAttributes ldNs and lnNs as shown in the excerpt of Table 14.

Table 14 – Excerpt of logical node name plate common data class (LPL)

cdcId = LPL, UML class name = LPL					
Attribute name	Attribute type	FC	TrgOp	Value/value range	PresCond
DataAttribute for configuration, description and extension					
...
ldNs	VisString255	EX		Logical device namespace, for example "IEC 61850-7-4:2007B".	MFlN0
InNs	VisString255	EX		Logical node namespace.	MOInNs

NOTE 1 The presence conditions in the last column are defined in IEC 61850-7-2.

The abstract type BasePrimitiveCDC groups attributes common to all common data classes. It contains the DataAttributes cdcName and dataNs as shown in the excerpt of Table 15.

Table 15 – Excerpt of the abstract common data class BasePrimitiveCDC

UML class name = BasePrimitiveCDC					
Attribute name	Attribute type	FC	TrgOp	Value/value range	PresCond
DataAttribute for configuration, description and extension					
...
cdcName	VisString255	EX		Name of the common data class.	O
dataNs	VisString255	EX		Data namespace.	MOdataNs

NOTE 2 The presence conditions in the last column are defined in IEC 61850-7-2.

13.5.2 Attribute for logical device namespace (ldNs)

The DataAttribute logical device name space ldNs shall be used to indicate the domain application name space used as the prime name space for the whole logical device, e.g. IEC 61850-7-4, IEC 61850-7-410, IEC 61850-7-420, etc.

The attribute ldNs shall be a DataAttribute of the name plate NamPIt of the logical node class LLN0. The attribute ldNs shall be as defined in the common data class LPL (logical node name plate) of IEC 61850-7-3. Its value shall be as specified in 13.4.

The attribute ldNs shall be available in every instance of the logical node class LLN0.

The ObjectReference for the DataAttribute ldNs shall be:

LDName/LLN0.NamPIt.ldNs

13.5.3 Attribute for logical device basic namespace (LLN0.NamPIt.InNs)

The DataAttribute logical node namespace LLN0.NamPIt.InNs shall be used to indicate the used basic namespace for a logical device. The version/revision of this basic namespace dictates the CDCs and their content that are implemented in the device. Examples can be found in Clauses J.7 and J.9. The attribute shall be available in an LLN0 instance if the used basic namespace differs from the logical device namespace. The attribute is optional in case the the used basic namespace and the logical device namespace would be the same.

The attribute InNs shall be a DataAttribute of the name plate NamPIt of a logical node LLN0. The attribute InNs shall be as defined in the common data class LPL (logical node name plate) of IEC 61850-7-3.

The ObjectReference for the DataAttribute InNs shall be:

LDName/LNN0.NamPlt.InNs

13.5.4 Attribute for logical node namespace (InNs)

The DataAttribute logical node namespace InNs shall be used to indicate the namespace that specifies the logical node class. The attribute shall be available in an LN instance if its value differs from the IdNs value. The attribute is optional in case the InNs and IdNs values would be the same.

The attribute InNs shall be a DataAttribute of the name plate NamPlt of a logical node. The attribute InNs shall be as defined in the common data class LPL (logical node name plate) of IEC 61850-7-3.

The ObjectReference for the DataAttribute InNs shall be:

LDName/LNName.NamPlt.InNs

The value of InNs shall be either:

- the standardized Namespace where the logical node class is defined
- the standardized Namespace where the logical node class has been extended as soon as the logical node instance is using the extension^{7,8}
- the private Namespace if the logical node has been defined outside the standard.

13.5.5 Attribute for data namespace (dataNs)

The DataAttribute data namespace dataNs shall be used to indicate the Namespace that specifies the usage of a compatible data object class. The attribute is optional for data object classes of the DomainLN class (e.g. NamPlt, Health, Mod)⁹. For all other data object classes, the attribute shall be available in an LN instance if its value differs from the InNs value. The attribute is optional in case the dataNs and InNs values would be the same.

The attribute dataNs shall be a DataAttribute of the data. The attribute dataNs shall be as defined in the common data classes of IEC 61850-7-3.

The ObjectReference for the DataAttribute dataNs shall be:

LDName/LNName.DataObjectName[.SubDataObjectName[. ...]].dataNs

13.5.6 Attribute for common data class namespace (cdcNs)

This attribute has been deprecated.

13.5.7 Attribute for naming a common data class name (cdcName)

The DataAttribute name of the common data class cdcName may be used for identifying a CDC in case of the extension of the model, e.g. in a technical report. An example can be found in Clause J.7.

⁷ The change of a presence condition in a product standard is considered as an extension.

⁸ Standardized extension of an enumeration is considered as an extension.

⁹ DomainLN is defined in IEC 61850-7-4.

13.6 Deprecation of namespaces

Namespaces have a life cycle, and the two main reasons may lead to deprecate namespaces are:

- The content of the considered namespace has moved to other parts, typically transitional namespace which content has migrated within new edition/version of domain or basic namespace(s), becomes de facto deprecated.
- A new edition/version of a given namespace is available and de facto makes the previous one deprecated

Deprecation of a given namespace means that it is not recommended anymore to build new namespaces upon the deprecated one.

These deprecated namespaces will still be made available for interoperability purpose; however no management will occur anymore, i.e. it won't be possible anymore to raise "tissues" on these namespaces. If any technical issues on a deprecated namespace, they may already have been corrected in newer edition/version, and if not, then the tissue must be based on the latest edition/version.

14 Common rules for new version of classes and for extension of object classes

14.1 General

This clause specifies rules for:

- Class version control.
- Extension of classes.
- The use of subsequent class versions (use of forward definition).

The rules description is supported by use case scenarios examples in Annex J.

The rules apply to the following classes: logical node (LN) classes, data classes, common data classes (CDC), enumerations and control block classes.

The rules are presented in the following order:

- Basic rules.
- Extensions made within private namespaces.
- Extensions made within product standard namespaces.
- Extensions made within basic and domain namespaces
- Extensions made within transitional namespaces.

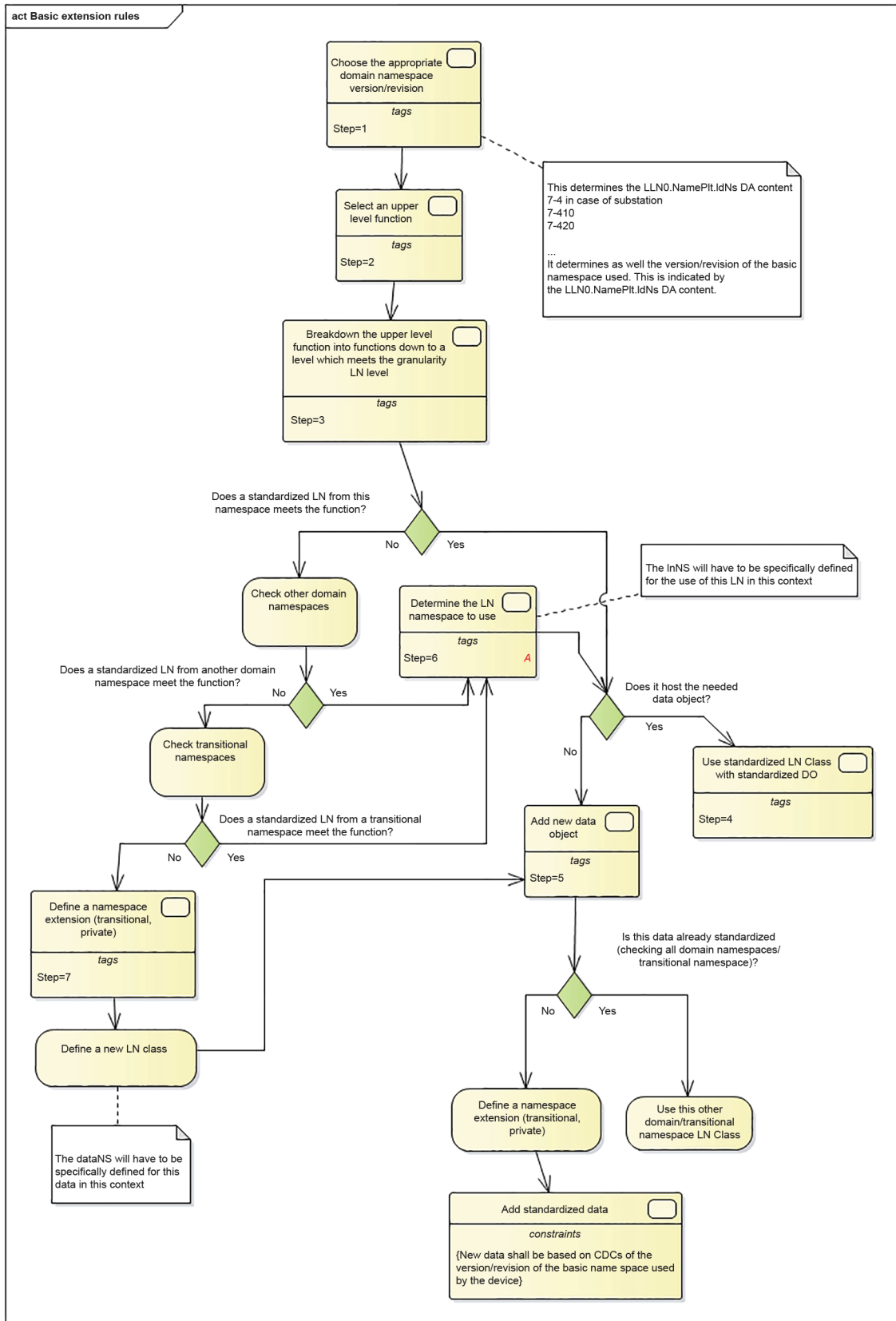
In the following, classes which are part of an owner namespace (e.g. IEC 61850-7-4) are called standardized classes.

14.2 Basic rules

14.2.1 General

14.2.1.1 Basic rules

The basic rules are depicted step by step in the diagram shown in Figure 79.



IEC

Figure 79 – Basic extension rules diagram

The first step is to choose a target domain namespace version/revision. This will determine the namespace of the logical device identified by the LLN0.NamePit.IdNs DA. It will dictate as well what version/revision of the basic namespaces are intended to be implemented in the

device. This basic namespace version/revision shall be specified by the attribute `NamPit.InNs` of the LN `LLN0`. See 13.5.3.

The second step is to select an upper level function to model. The third step is to decompose the required application function to a degree of granularity of the existing logical node classes of the application domain, e.g. substation automation in case of the IEC 61850 series that are listed in IEC 61850-7-4. From there, the rules described in the following subclauses shall be applied.

14.2.1.2 Use of standardized LN classes with standardized data objects classes (step 4)

When a standardized LN class in the chosen domain namespace fulfils the function, it shall be used and the following rules shall be applied:

- a) If there is any logical node class which fits the function to be modelled, an instance of this logical node shall be used with all its mandatory data (indicated by the presence conditions in the LN table). The rules of a unique instantiation can be found in IEC 61850-7-2.
- b) If, in addition to the mandatory data, there are also optional data (indicated by the presence conditions in the LN table), which fit the function to be modelled, these optional data shall be used.
- c) data instances shall be used with all their mandatory data attributes (indicated by the presence conditions in the CDC table). If there are also optional data attributes (indicated by the presence conditions in the CDC table), which fit the function to be modelled, these optional data attributes shall be used.
- d) If there is the same data which need to be instantiated several times, additional data with number extensions shall be used according to the rules for specialisation of data by use of number extensions, see 14.8.
- e) If there are dedicated versions of the function to be modelled with the same basic data (for example ground, phase, zone A, zone B, etc.), different instances of this logical node class shall be used, see 14.7.
- f) Subsequent LN class definitions (forward definitions¹⁰) may be used provided that the version/revision of the CDCs is the same as the one of the basic namespace used (see Clause J.5).

14.2.1.3 Extending standardized LN classes with new data objects classes (step 5)

When extending standardized LN classes with new data objects classes, the following rules shall be applied:

- a) If a data object class with the same semantic, within the context of the LN class, is already standardized in another domain/transitional namespace, this other domain/transitional namespace LN class shall be used (See clause J.4).
- b) If not, a new data object class based on CDCs of the version/revision of the basic namespace used by the device shall be created in a transitional/private namespace.

14.2.1.4 Use of standardized LN classes from other namespaces (step 6)

When none of the standardized LN class in the chosen domain namespace fulfils the function and an LN class from another domain/transitional namespace does, it shall be used.

¹⁰ A forward definition of a class is a class that has been defined in a next version/revision of the domain application namespace used.

14.2.1.5 Extending a domain namespace with new LN classes (step 7)

When extending a domain namespace with new LN classes, the new LN classes shall be created in a transitional/private namespace.

The new LN classes inherit all data object classes from the DomainLN class. For these data objects the presence of dataNs is optional, when present it shall have a same value as the applicable LLN0.NamPit.InNs¹¹.

14.2.2 Use of optional information

The presence condition of many object classes in IEC 61850 is defined as optional. E.g. some LNs like MMXU even have no specific data objects. This is in part due to the general agreement about modelling granularity. In the MMXU example, instead of having tens of LNs related to operative or statistical measurements, there is only one LN with tens of optional data objects that are used depending on the required measurement. This raises the question how to use the optional information.

Whether it is in the context of an end user or a system/device manufacturer, data objects or data attributes shall be used in order to fulfill a well specified functionality. Only the required objects should be used. In the case of Basic Application Profiles (IEC 61850-7-6 BAP), since they describe profiles dedicated to a group of users, all objects mentioned as mandatory are considered as such, even though the standard does not.

Devices (servers and clients) should offer flexibility regarding the use of optional information in reports. Servers shall allow for flexible configuration of datasets at the data level (FCD). Clients should be flexible enough to receive attributes (FCDA) whose associated functionalities are not supported by the client. One may refer to IEC 61850-7-2 for an explanation about reporting and data set members' inclusion.

14.2.3 General rules for enumerations types and enumeration-based data objects and attributes

The rules from IEC 61850-7-2 shall be followed.

14.3 Rules for extensions within private namespace

14.3.1 Rules for LN classes

Extensions to standardized LN classes may be made by third parties. However, third parties cannot change the semantics of a LN and therefore cannot change the name spacename when making extensions. To not jeopardize interoperability, the rules for LN extensions are the following.

- a) The new data object class name shall follow the naming conventions defined in IEC 61850-7-2 and dataNs shall be mandatory and be a private namespace.
- b) The new data object class shall be assigned to one and only one of the common data classes defined in IEC 61850-7-3.
- c) Standardized data object classes may be used. In such a case (see Clause J.3).
 - The semantic shall be the one of the standardized data object class.
 - dataNs shall be mandatory and be a private namespace.
 - The CDC shall be the same as the standardized data object class one.

¹¹ Refer to IEC 61850-7-4 for a definition of the DomainLN class.

- d) Subsequent standardized data object class definitions (forward definitions¹²) may be used. The dataNs attribute values shall be set accordingly. In such a case, (see Clause J.4):
 - The semantic and type shall be the one of the standardized data object class.
 - dataNs shall be mandatory and be the one of the standardized data object class.
- e) The description of the new data object class shall be added to the IEC documentation of the provided specific system or the customer specific project.

Defining private data object classes may have drawbacks and one should be aware of the following remark.

As soon as a standardized namespace introduces a private data object class, then as soon as the device is claimed to be conformant to the new version of the namespace, then:

- If the used CDC as well as the semantic of the DO match the standard, the device must comply with the new standardized data object class and the dataNs shall be the standardized one.
- If the semantic of the used CDC does not match, then the dataNs shall remain a private one and the private data object class shall be renamed.

14.3.2 New LN classes

Third parties may expand the owner LN class set by defining new LNs outside the owner's namespace. In such a case, the rules are:

- a) InNs shall be mandatory and be a private namespace.
- b) The new LN class shall contain the data object class Beh which shall follow the LD hierarchy definition.
- c) Standardized data object classes can be reused (See Figure I.1). In such a case:
 - The semantic shall be the one of the standardized data object class.
 - The CDC shall be the same as the standardized data object class one.
 - The presence of dataNs is optional, when present it shall have a private namespace value.

The description of the new LN class shall be added to the IEC documentation of the provided specific system or the customer specific project. Defining private LN classes may have drawbacks and one should be aware of the following remarks.

As soon as a standardized namespace introduces the functionality of the private LN class, then as soon as the device is claimed to be conformant to the new version of the namespace, then:

- a) If the functionality of the private LN class match the standard, the device shall comply with the new standardized LN class and the dataNs shall be the standardized one (previous private LN instances may remain for concerns of backward compatibility issues within the system).
- b) If the functionality of the private LN class does not match, then the InNs shall remain a private one and the private LN class shall be renamed.

14.3.3 Rules for Common Data Classes

Starting from Edition 2 (2011) of this standard, new versions of standardized common data classes shall only be made by the owner of the name space. As a consequence, private extension of an existing CDC shall be prohibited.

¹² A forward definition of a class is a class that has been defined in a next version/revision of the domain application namespace used.

14.3.4 Rules for enumeration types and enumeration-based data objects and attributes

14.3.4.1 Private extensions for standardized enumeration-based data attributes

Private extensions following general rules defined in 14.2.3 are allowed but may however lead to interoperability issues.

14.3.4.2 Private extensions for standardized enumeration-based data objects

Private extensions following general rules defined in 14.2.3 are allowed but may however lead to interoperability issues.

14.3.4.3 Private extensions for private enumeration-based data objects

Private data objects may have private enumerations.

14.4 Extensions made within product standard namespaces

The IEC 61850 series is a set of horizontal standards that covers several application domains. From that point of view, product family standards can be seen as vertical standards implementing particular parts of the IEC 61850 series.

TCs responsible for product family standards may make specific extensions to other namespaces (see Figure 79). One can refer to IEC 61850-1-2 for further details about rules for extensions within product standard namespace. An example is given in Clause J.8.

14.5 Extensions made within transitional namespaces

Transitional namespaces (the IEC TR 61850-90-XXX series) are intended to make proposals for extensions to other namespaces (see Figure 79). These extensions may then be integrated in the affected parts of the IEC 61850 series. An example is given in Clause J.6.

Transitional namespaces are allowed under specific process to create new CDC. An example is given in Clause J.7. Transitional namespaces are not allowed to extend existing CDC. Refer to IEC 61850-1-2 for guidelines about extensions within transitional namespaces.

NOTE These extensions are not subject to conformance testing.

14.6 Extensions made within basic and domain namespaces

Domain namespaces contain definitions applicable to particular domains such as substation automation or distributed energy resources. Any definition in a domain namespace may be reused in another domain namespace. So before making an extension, a domain namespace owner should check that the extension doesn't already exist in another domain namespace. One can refer to IEC 61850-1-2 for further details about rules for extensions within basic and domain namespaces.

NOTE These extensions are subject to conformance testing.

14.7 Multiple instances of LN classes for dedicated and complex functions¹³

14.7.1 Example for time overcurrent

Logical node class name: PTOC (time overcurrent)		
LN instance name (with LN-Prefix)	Meaning	Meaning of "Start value" StrVal
GndPTOC1	Ground fault detection	"Ground Start Value"
PhsPTOC1	Phase fault detection	"Phase Start Value"

The semantics of the different instances may be given in the description attribute "d" of data NamPIt (name plate).

14.7.2 Example for PDIS

Logical node class name: PDIS (distance)	
LN instance name (without LN-Prefix)	Meaning
PDIS1	Zone 1 of the distance protection
PDIS2	Zone 2 of the distance protection
PDIS3	Zone 3 of the distance protection
Etc.	Etc.

The semantics of the different instances may be given in the description attribute "d" of data NamPIt (name plate).

14.7.3 Example for power transformer

Logical node class name: YPTR (power transformer)	
LN instance name (without LN-Prefix)	Meaning
YPTR1	Transformer unit phase L1
YPTR2	Transformer unit phase L2
YPTR3	Transformer unit phase L3

The semantics of the different instances may be given in the description attribute "d" of data NamPIt (name plate).

14.7.4 Example for auxiliary network

Logical node class name: ZAXN (auxiliary network)	
LN instance name (without LN-Prefix)	Meaning
ZAXN1	220 V DC
ZAXN2	60 V DC
ZAXN3	380 V DC

The semantics of the different instances may be given in the description attribute "d" of data NamPIt (name plate).

¹³ The examples supplied in this clause are for informative purpose only and are therefore non normative.

14.8 Specialisation of data by use of number extensions

Standardised data names in logical nodes provide a unique identification. If the same data object (i.e. data with the same semantics) are needed several times as defined, additional data objects with number extensions shall be used. The rules for number extensions shall follow the naming conventions defined in IEC 61850-7-2 and be as follows:

- the number extension usage shall only be defined by the owner of the data namespace. This shall be done by fixing the presence condition (PresCond) of the data object to Omulti,
- data objects without presence condition (PresCond) equal to Omulti shall not be extended by third parties,
- data objects with presence condition (PresCond) equal to Omulti can be extended. Number extensions may be ordered or not (1,2,3,4, or, 1,2,25, 19),
- even if only one instance of an extendable data is present in an LN, it shall have a number extension.

Examples are given hereafter:

Logical node class name: PSCH (Protection scheme)	
Data name: RxTr1 (teleprotection direct trip signal)	
RxTr1	teleprotection direct trip signal 1
RxTr2	teleprotection direct trip signal 2
RxTr3	teleprotection direct trip signal 3

14.9 Examples for new LNs

New LN “Automatic door entrance control”

1 st character Logical node group indication	2 nd character	3 rd character	4 th character	New LN	Value of ADEC.NamPit.InNs
A for “Automatic Control”	Door	Entrance	Control	ADEC “Automatic door entrance control”	myVendor1Extension

New LN “Fire protection”

1 st character Logical node group indication	2 nd character	3 rd character	4 th character	New LN	Value of ZFPT.NamPit.InNs
Z for “Further equipment”	Fire	Protection	Transformer	ZFPT “Fire protection of a power transformer”	myVendor2Extension

14.10 Example for new Data

New Data “Colour of Transformer Oil”

Data name	Attribute type (CDC)	Value of ColrTOil.dataNs
ColrTOil	INS	myVendor1Extension

15 Compatibility between different versions of the standard

Due mainly to evolution, different versions of this standard will be published. All these versions shall be backward and forward compatible between each other. In for any reason this is not possible, general requirements and recommendations regarding compatibility issues are provided in Annex K.

Annex A (informative)

Overview of logical nodes and data

A.1 Compatible logical node classes and data classes (IEC 61850-7-4)

A.1.1 List of LN groups (IEC 61850-7-4)

A list of all groups of logical nodes is contained in Table 1 of this part of IEC 61850.

A.1.2 LN classes (IEC 61850-7-4)

An excerpt of groups of logical nodes is contained in 5.4 of this part of IEC 61850.

A.1.3 Data object classes (IEC 61850-7-4)

Data object classes are defined in each logical node in IEC 61850-7-4.

The data object names are composed using standardised abbreviations listed in a table at the beginning of IEC 61850-7-4, for example:

Term	Description
A	Current
Acs	Access
Acu	Acoustic
Age	Ageing
Alm	Alarm
Amp	Current non phase related
An	Analogue
Ang	Angle
...	...

These abbreviations should be used in the creation of new data names.

The common data class to be used with a specific DATA item is defined in the logical nodes.

A.2 Common data class specifications (IEC 61850-7-3)

Table A.2 shows an excerpt of the list of the common data classes defined in IEC 61850-7-3. All common data classes are used by one or the other logical node.

Table A.2 – List of common data classes (excerpt)

Common data class specifications for status information
Single point status (SPS)
Double point status (DPS)
Integer status (INS)
Enumerated status (ENS)
Protection activation information (ACT)
Directional protection activation information (ACD)
Security violation counting (SEC)

Common data class specifications for status information
Binary counter reading (BCR)
Histogram (HST)
...
Common data class specifications for measurand information
Measured value (MV)
Complex measured value (CMV)
Sampled value (SAV)
Phase to ground/neutral related measured values of a three-phase system (WYE)
Phase to phase related measured values of a three-phase system (DEL)
Sequence (SEQ)
Harmonic value for WYE (HWYE)
...
Common data class specifications for controllable information
Controllable single point (SPC)
Controllable double point (DPC)
Controllable integer status (INC)
Controllable enumerated status (ENC)
Binary controlled step position information (BSC)
Integer controlled step position information (ISC)
...
Common data class specifications for status settings
Single point setting (SPG)
Integer status setting (ING)
Enumerated status setting (ENG)
Object reference setting group (ORG)
...
Common data class specifications for analogue settings
Analogue setting (ASG)
Setting curve (CURVE)
...
Common data class specifications for description information
Device name plate (DPL)
Logical node name plate (LPL)
...

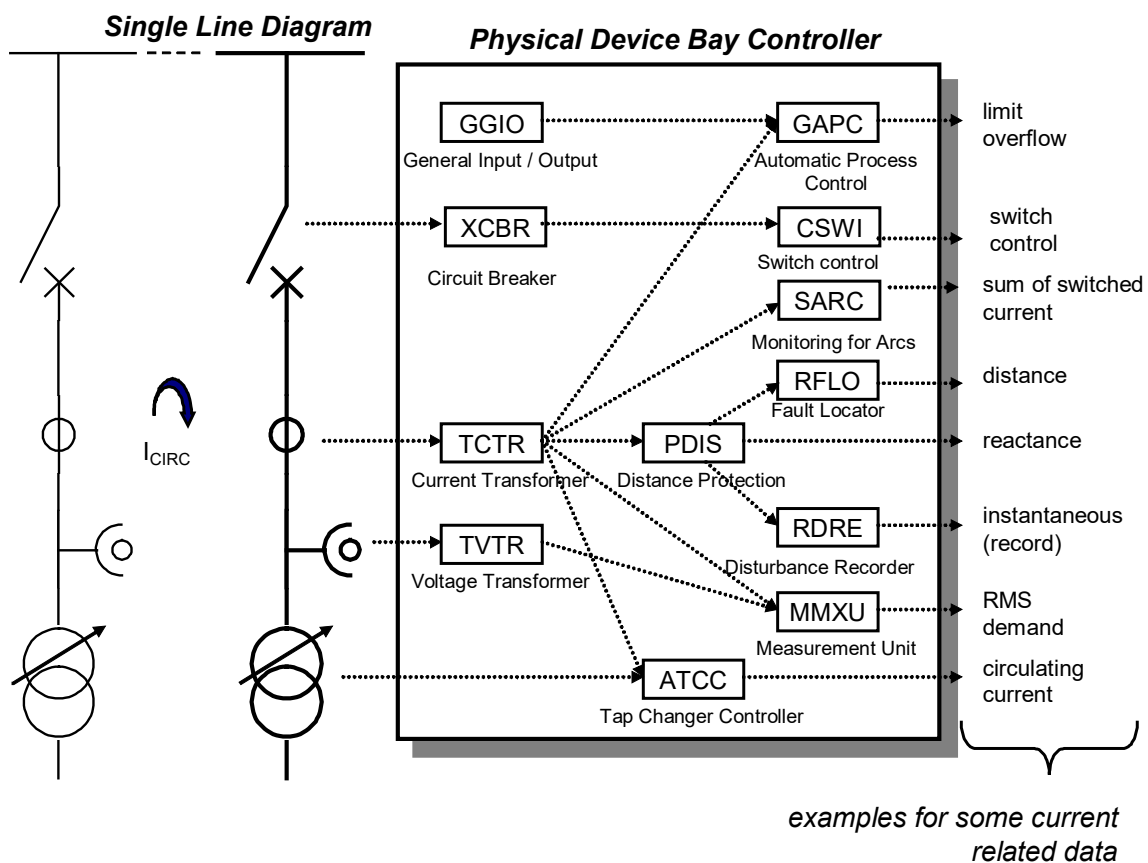
Annex B (informative)

Allocation of data to logical nodes

Figure B.1 illustrates an example of the assignment of data to logical nodes.

A data is assigned to that logical node that produces or consumes the values of this data object, this means for example:

- data consisting of instantaneous values of current and voltage are assigned to the logical nodes “current transformer” and “voltage transformer” respectively;
- data consisting of calculated values of current and voltage, for example root-mean-square r.m.s., are assigned to the logical node “measurement unit”;
- data consisting of voltage and step position are assigned to the logical node “tap changer controller”. Same holds for the tap change commands;
- data consisting of a (fault) impedance Z are assigned to the logical node “distance protection”. The same applies to the protection trip.



IEC 1480/11

Figure B.1 – Example for control and protection LNs combined in one physical device

In any case where a compatible data has been defined in the standard for a specific application, this compatible data shall be used instead of defining a new one as specified in the rules for extension in Clause 14.

A second example application is shown in Figure B.2. The merging unit receives the current and voltage values directly from the instrument transformers. This unit may exist integrated per instrument transformer. These implementations are outside the scope of the standard. The sources of the samples are always instances of the LN classes TVTR and TCTR. In the example with the merging unit, the samples of all three phases and the neutral are collected and send out as multicast messages. Several applications receive the sampled values.

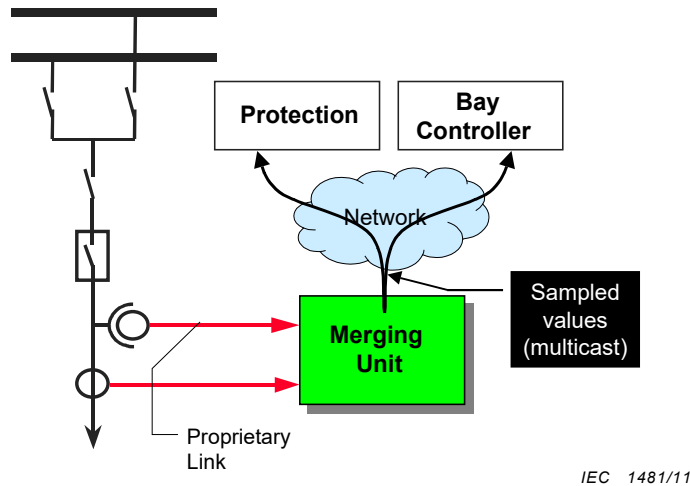


Figure B.2 – Merging unit and sampled value exchange (topology)

The merging unit is modelled as a single logical device named “MergingUnit” (see Figure B.3).

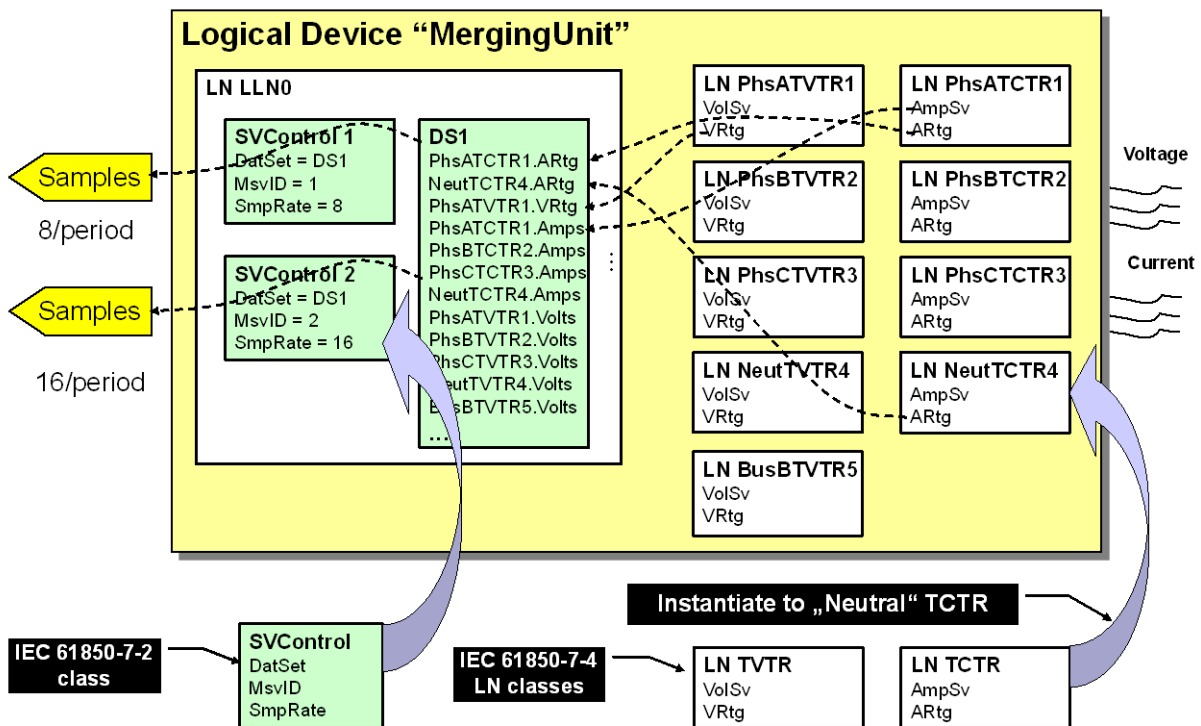


Figure B.3 – Merging unit and sampled value exchange (data)

The current and voltage values are received at the right-hand side. The three-phase values for current and voltage are modelled in the following logical nodes:

- Current transformer – TCTR class in IEC 61850-7-4 instantiated for three phases and neutral:
PhsATCTR1, PhsBTCTR2, PhsCTCTR3, NeutTCTR4,
- Voltage transformer – TVTR class in IEC 61850-7-4 instantiated for three phases, neutral, and bus bar:
PhsATVTR1, PhsBTVTR2, PhsCTVTR3, NeutTVTR4, BusBTVTR5,

The sampled values (AmpSv and VolSv) and the corresponding ratings are used in the example. These data are referenced by the data set “DS1”.

Two instances of the sampled value control blocks (SVControl1 and SVControl2) are defined to control the exchange of the samples. The two control blocks just realise two different sample rates (8 and 16 samples per nominal period – 400/800 samples per second in the case of a 50 Hertz system).

Annex C (informative)

Use of the substation configuration language (SCL)

C.1 General

This annex explains only the application of the SCL to define the use of optional definitions contained in class definitions in IEC 61850-7-4 and IEC 61850-7-3.

C.2 SCL and options in logical nodes

Figure C.1 shows the logical node class XCBR as it is defined in IEC 61850-7-4. There are several data items defined as being mandatory (M); other data are defined as being optional (O).

A logical node (XCBR) of a device model is specified with a SCL file. By definition, all mandatory data defined in the class defined in IEC 61850-7-4 are used by the logical node in the device model.

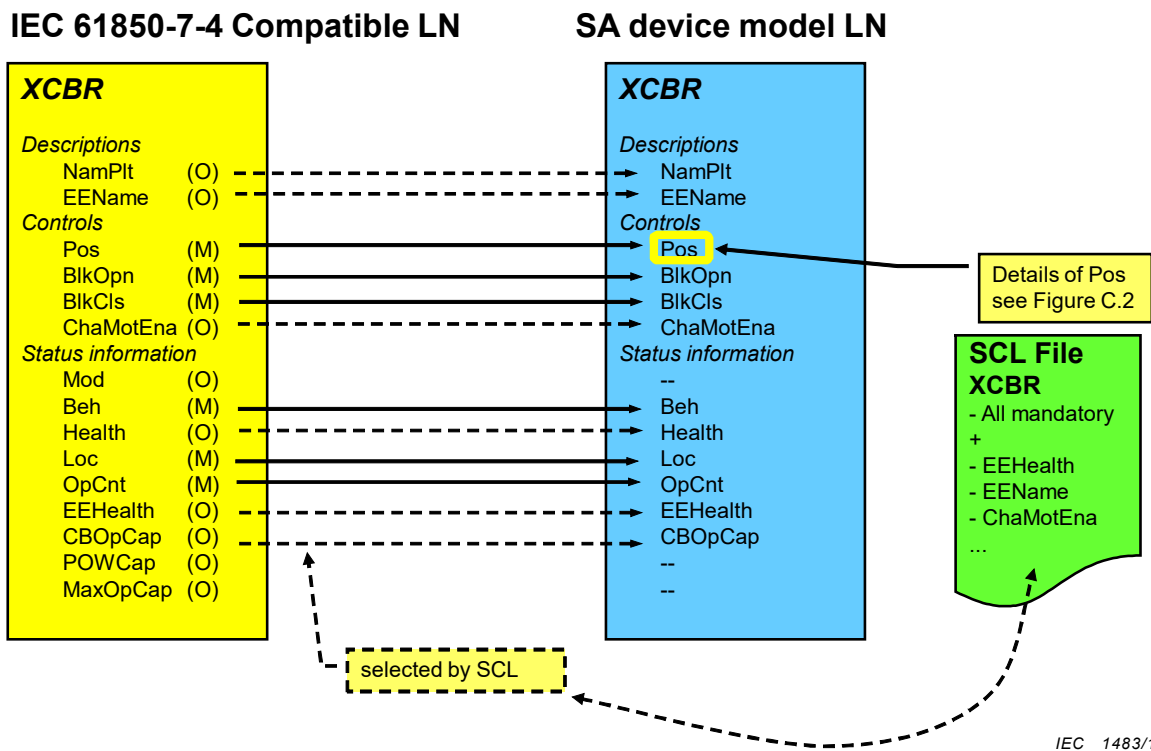


Figure C.1 – Application of SCL for LNs (conceptual)

The SCL needs to list all data to be used in the device model. Three optional data items are selected in the example (EEHealth, EEName, and ChaMotEna).

The SCL also needs to list the optional data attributes of each data selected. The marked data Pos is detailed in Clause C.3.

C.3 SCL and options in data

At the logical node level, the SCL shall list the names of the mandatory and optional data. The SCL for the data requires the list of mandatory and optional data attributes as well as the initialisation (configuration) values for several data attributes.

The SCL file in Figure C.2 shows which optional data attributes are selected. In addition, the SCL file assigns values to three data attributes.

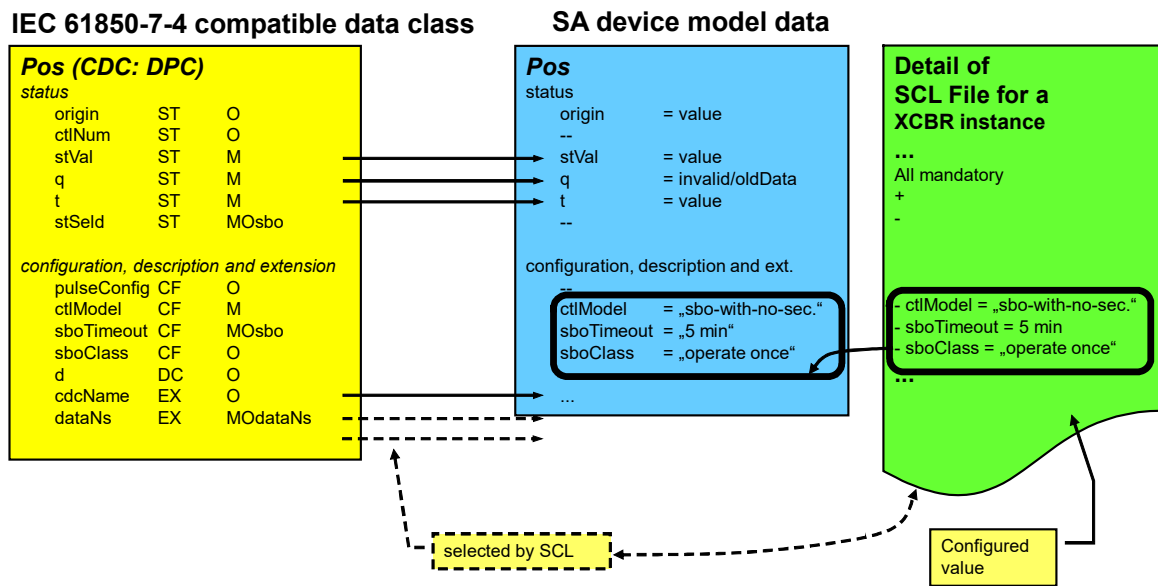


Figure C.2 – Application of SCL for data (conceptual)

The configured values for `ctlModel`, `sboTimeout`, and `sboClass` become effective as soon as the real device has been configured. The values may be overwritten (if the device allows overwriting these values at all) by a service request from a specific client.

Annex D
(xxx)

This page is intentionally left blank.

Annex E (informative)

Relation between logical nodes and PICOMs

IEC 61850-5 describes functions in a substation automation system that are divided into subfunctions called logical nodes. The content of the exchanged data between the LNs are (in IEC 61850-5) called PICOMs (pieces of communication), see Figure E.1. This view is independent of any model that is used to define the semantic and syntax of exchanged data – such as the client/server model.

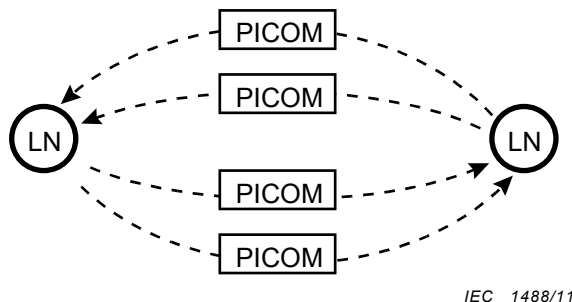


Figure E.1 – Exchanged data between subfunctions (logical nodes)

In the client/server model, there are defined services that determine the semantics and syntax of exchanged data. In this sense, the exchanged data is called protocol data unit (PDU) that determines the “bits on the wire”. The content and the semantics of the exchanged data is determined by the objects inside the server. In this model, the logical nodes are objects. Their subcomponents – the data objects – comprise all the process information that is related to the content of the PICOMs, see Figure E.2.

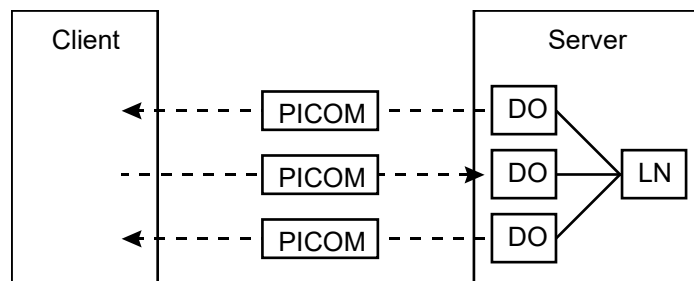


Figure E.2 – Relationship between PICOMS and client/server model

Since the logical nodes are objects in the client/server model, the subfunctions (LNs) inside a client are not of interest to describe the communication with the server.

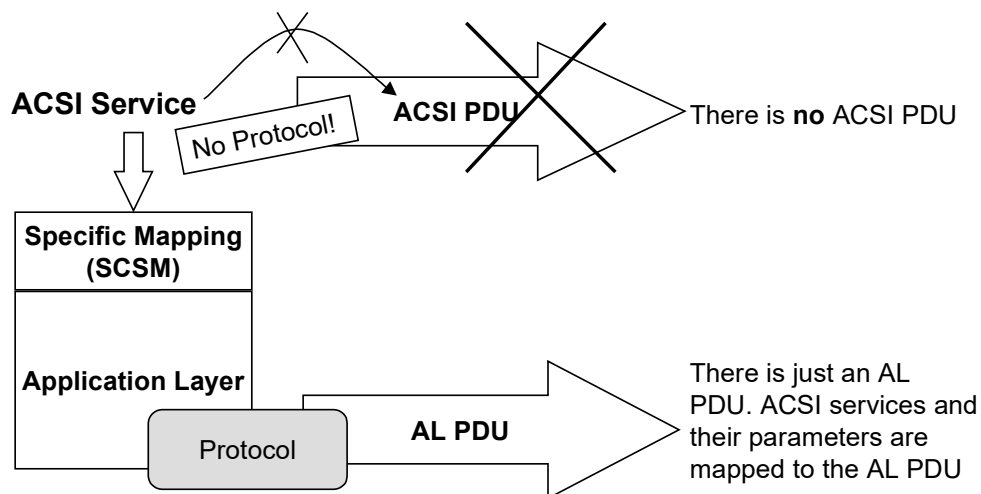
One PDU may comprise the content of several data – therefore the content of several PICOMs.

Annex F (informative)

Mapping the ACSI to real communication systems

F.1 General

Figure F.1 depicts the relation of the ACSI to an underlying application layer. The ACSI does not define concrete ACSI messages. The ACSI services are mapped to a series of one or more application layer messages (AL PDU – protocol data unit) of the underlying application layer.



IEC 1490/11

PDU: Protocol data unit (encoded message containing the service parameter, etc.).

Figure F.1 – ACSI mapping to an application layer

The mapping of ACSI services to specific application layer messages is beyond the scope of IEC 61850-7-2. This mapping is specified by a specific communication service mapping (SCSM) in the IEC 61850-8-x and the IEC 61850-9-x series.

NOTE 1 This mapping allows the ACSI to be applied to different application layers. As these application layers provide different features, the mapping within the SCSM may be simple or complex, and more or less efficient.

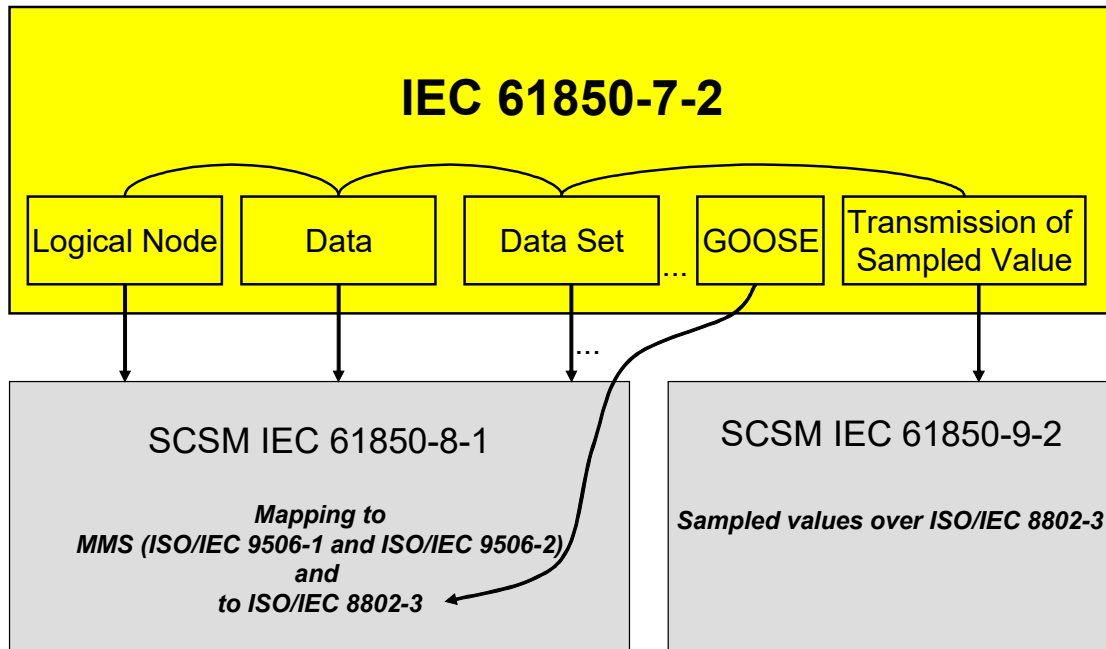
IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2 define abstract information and service models for the application domain substation. Even so, the IEC 61850 series in general allows discrete devices to share data and services. For this to occur, the devices shall agree on the concrete form of the services and data that will be exchanged.

The form of the service and data is of no consequence to the transport, network, and media protocols, i.e. to the lower layers of the communication stack and they are invariant to it. Conversely, the application that is sending and receiving data has no real procedure describing how this is achieved and it is therefore largely invariant of the mechanisms used.

This separation of roles is important as it allows many different technologies to be employed in a relatively transparent manner. As consequence, these lower layers may be exchanged. For example,

- networks with different types of physical media may be used;
- more than one application layer protocol may exist and use the same physical network and protocols.

Standardised mappings of the abstract services to different communication stacks are defined in the IEC 61850-8-x and the IEC 61850-9-x series, so that common utility functions will be performed consistently across all field devices independently of the underlying communication systems. Figure F.2 summarises the mappings defined in IEC 61850-8-1 and IEC 61850-9-2.



IEC 1491/11

Figure F.2 – ACSI mappings (conceptual)

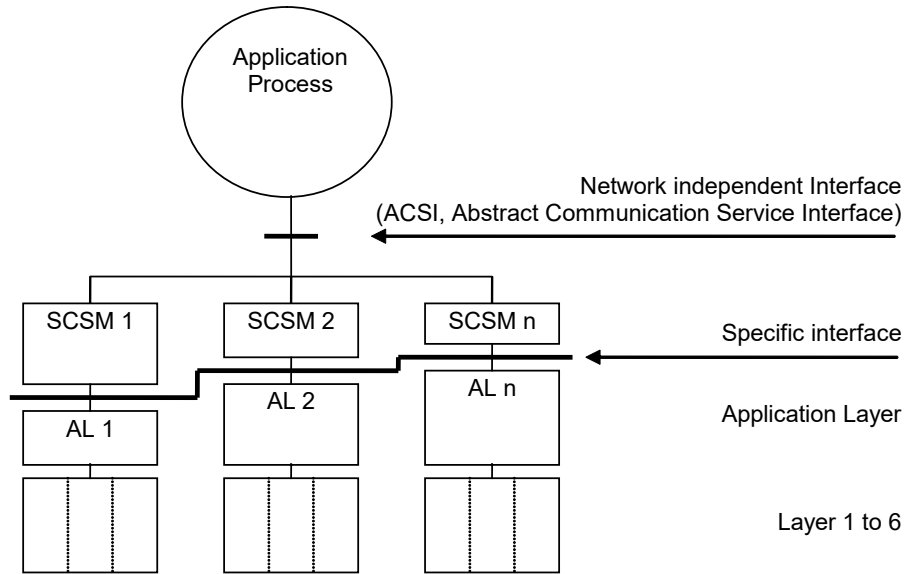
All but GOOSE and transmission of sampled values are mapped to MMS, TCP/IP, and ISO/IEC 8802-3. GOOSE is mapped directly to ISO/IEC 8802-3. The transmission of sampled values is mapped in IEC 61850-9-2.

The specific communication service mapping defines how the services and the models (server, logical devices, logical nodes, data, data sets, report controls, log controls, setting groups, etc.) are implemented using a specific communication stack, i.e. a complete profile. The mappings and the used application layer define the syntax (concrete encoding) for the data exchanged over the network.

NOTE 2 The concept of the SCSM has been introduced to be independent from communication stacks including application protocols. One objective of the IEC 61850 series is the interoperability of devices. This requires that all communicating devices use the same communication stack. Therefore, the goal of this independence is not to have many mappings in parallel, but to be able to follow the state of the art in communication technology.

According to Figure F.3, the SCSM maps the abstract communication services, objects and parameters to the specific application layers. These application layers provide the concrete coding. Depending on the technology of the communication network, these mappings may have different complexities, and some ACSI services may not be supported directly in all mappings, but equivalent services shall be provided (see example below). An application layer may use one or more stacks (layer 1 to 6).

EXAMPLE The ACSI service “GetDataSetValues” may have different mappings for different application layers (AL). For example, a specific AL may support this service directly while another AL provides “Get of single data” only. In the last case the mapping has to issue several “Get of single data”.



IEC 1492/11

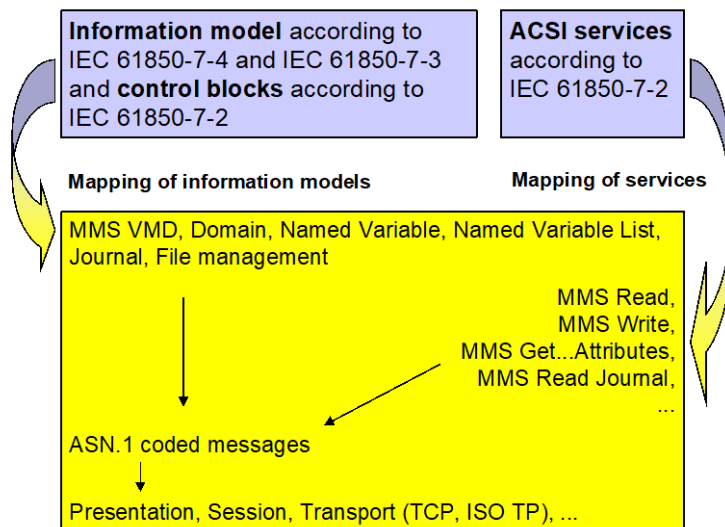
Figure F.3 – ACSI mapping to communication stacks/profiles

F.1.1 Mapping example (IEC 61850-8-1)

The information models (logical device, logical node, data, and data attributes) are defined in an abstract way in IEC 61850-7-4 and IEC 61850-7-3. In addition, the service models are defined as abstract services (ACSI – abstract communication service interface) defined in IEC 61850-7-2.

NOTE The names of logical nodes, data and data attributes are used as they are defined. The name XCBR is kept as the name XCBR. Mappings that do not support names in their protocols may map the names to unique numbers.

The abstract models of IEC 61850-7-4, IEC 61850-7-3 and IEC 61850-7-2 need to be mapped to an application layer (see Figure F.4).



IEC 1493/11

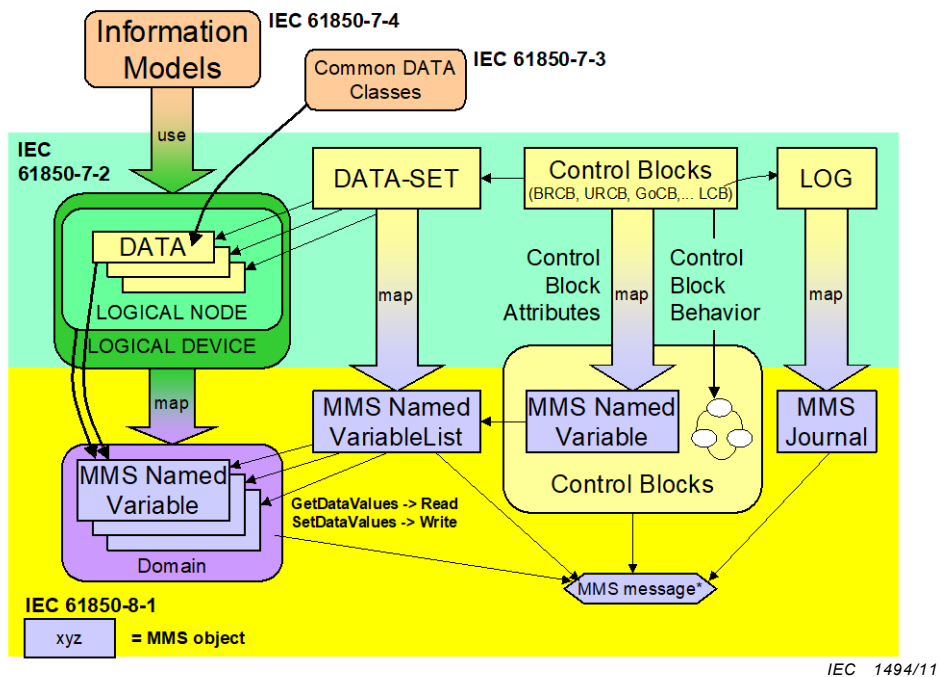
Figure F.4 – Mapping to MMS (conceptual)

The information model and the various control blocks are mapped in this example to the manufacturing message specification (MMS), i.e., to the virtual manufacturing device (VMD), domain, named variable, named variable list, journal, and file management. The services are mapped to the corresponding services of the MMS classes.

A more detailed mapping is shown in Figure F.5. The abstract information models defined in IEC 61850-7-4, IEC 61850-7-3 and IEC 61850-7-2 map as follows:

What to be mapped?	Maps to
Logical device (contains logical nodes); IEC 61850-7-2	MMS domain
Logical nodes (contains data); IEC 61850-7-4	MMS named variable
Data; IEC 61850-7-4	MMS named variable (and structured components of the named variable representing the “logical node data”)
Data attribute; IEC 61850-7-3	MMS named variable (and structured components of the named variable representing the “data”)
Data set: IEC 61850-7-2	MMS named variable list
Control blocks (attributes); IEC 61850-7-2	MMS named variable
Control blocks (behaviour); IEC 61850-7-2	Needs to be programmed as defined in IEC 61850-7-2
Log; IEC 61850-7-2	MMS journal

The messages carrying the information are mapped to MMS messages except for the GOOSE and SV messages.



* GOOSE/SMV messages map directly to ISO/IEC 8802-3.

Figure F.5 – Mapping approach

The details of the mapping to the MMS named variable are sketched in Figure F.6.

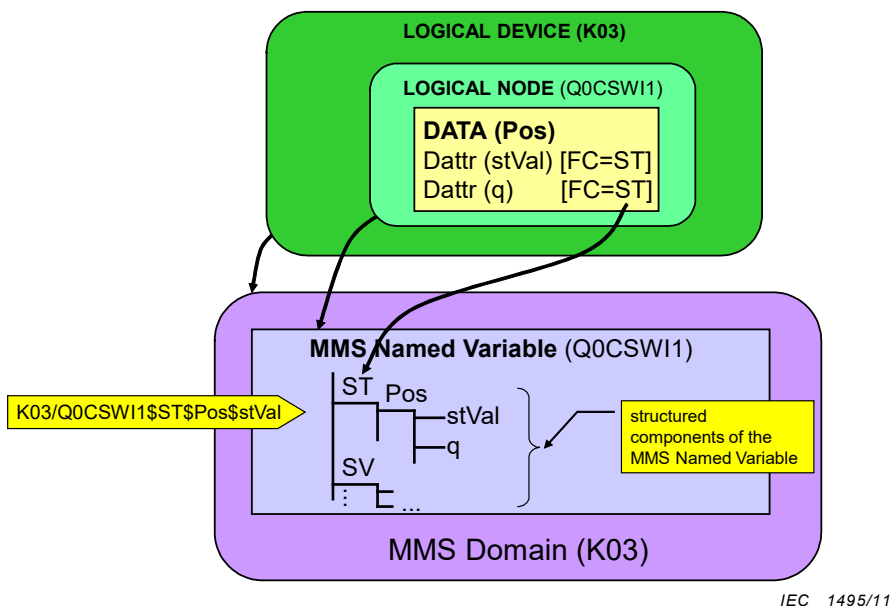


Figure F.6 – Mapping detail of mapping to a MMS named variable

The MMS Domain (with the name K03) contains named variables. The named variable shown in Figure F.7 has the name “Q0CSW11”. The components of this named variable are constructed by selecting all data attributes with the same functional constraint (FC), for example, the value FC=ST (all status data attributes). The first component of the named variable has the component name “ST”. The DATA (for example, Pos) are placed at the next nesting level. The data attributes (for example, stVal, q, t, etc.) are at the next level below. The dots “.” in the hierarchical name have been replaced by “\$” in the MMS mapping.

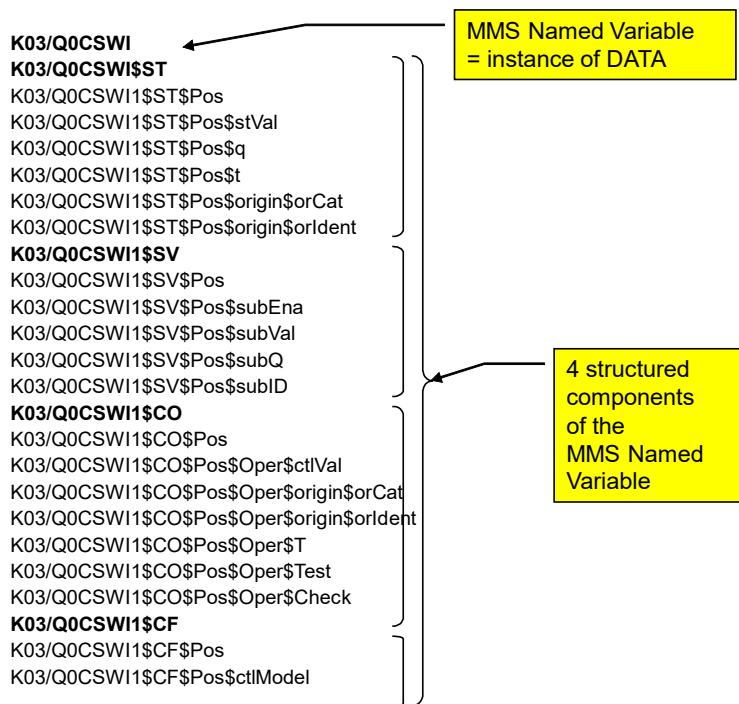
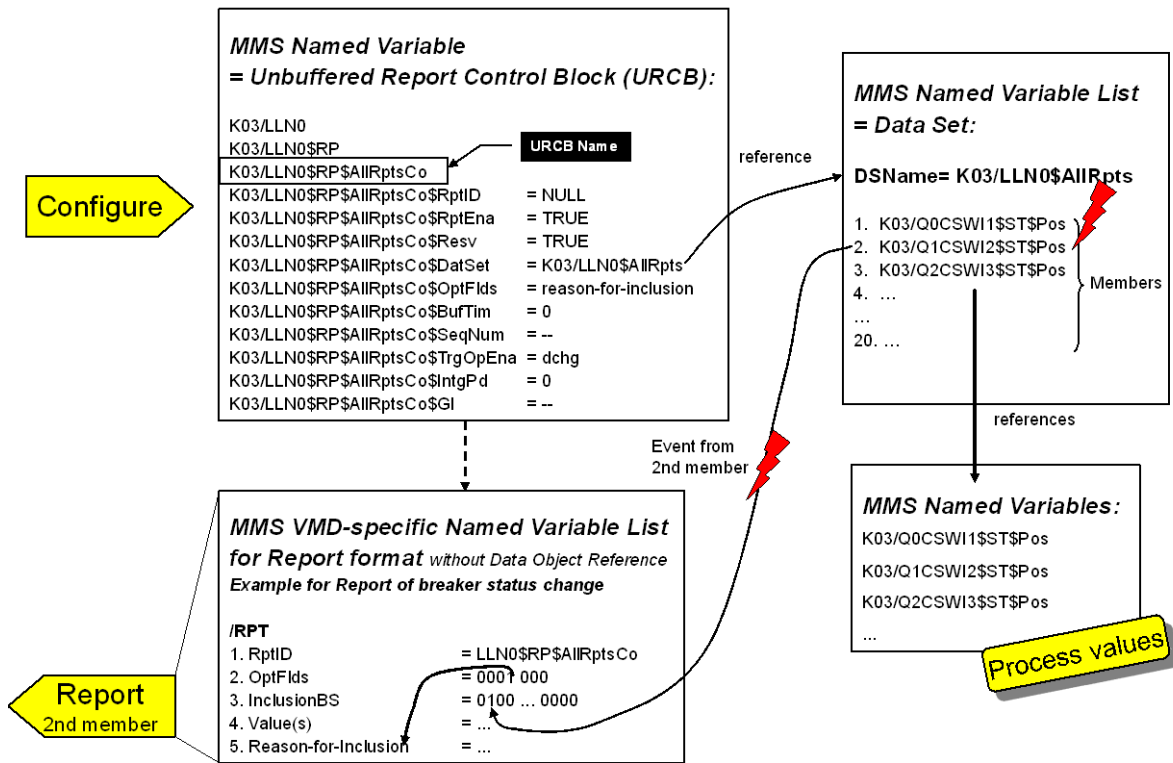


Figure F.7 – Example of MMS named variable (process values)

The process values (from the position) of several switch controllers Q0CSWI1\$Pos, Q1CSWI2\$Pos, and Q2CSWI3\$Pos are mapped to MMS named variables (see right lower corner of Figure F.8). The position information is grouped by the named variable list (data set) with the name K03/LLN0\$AIIrpts. The attributes of the unbuffered report control block are mapped to the MMS named variable K03/LLN0\$RP\$AIIrpts. The components of this named variable can be written (configured). The control block references the data set.



IEC 1497/11

Figure F.8 – Use of MMS named variables and named variable list

A change in one of the members of the data set (for example, in member 2) issues sending a report with the status of the position of Q1CSWI2. The report message is generated using another MMS named variable list (left lower corner). The report will be sent immediately.

The report is mapped to an MMS information report (see Figure F.9). The figure shows the concrete encoding according to ASN.1 BER (the basic encoding rule for abstract syntax notation number one – ISO 8825).

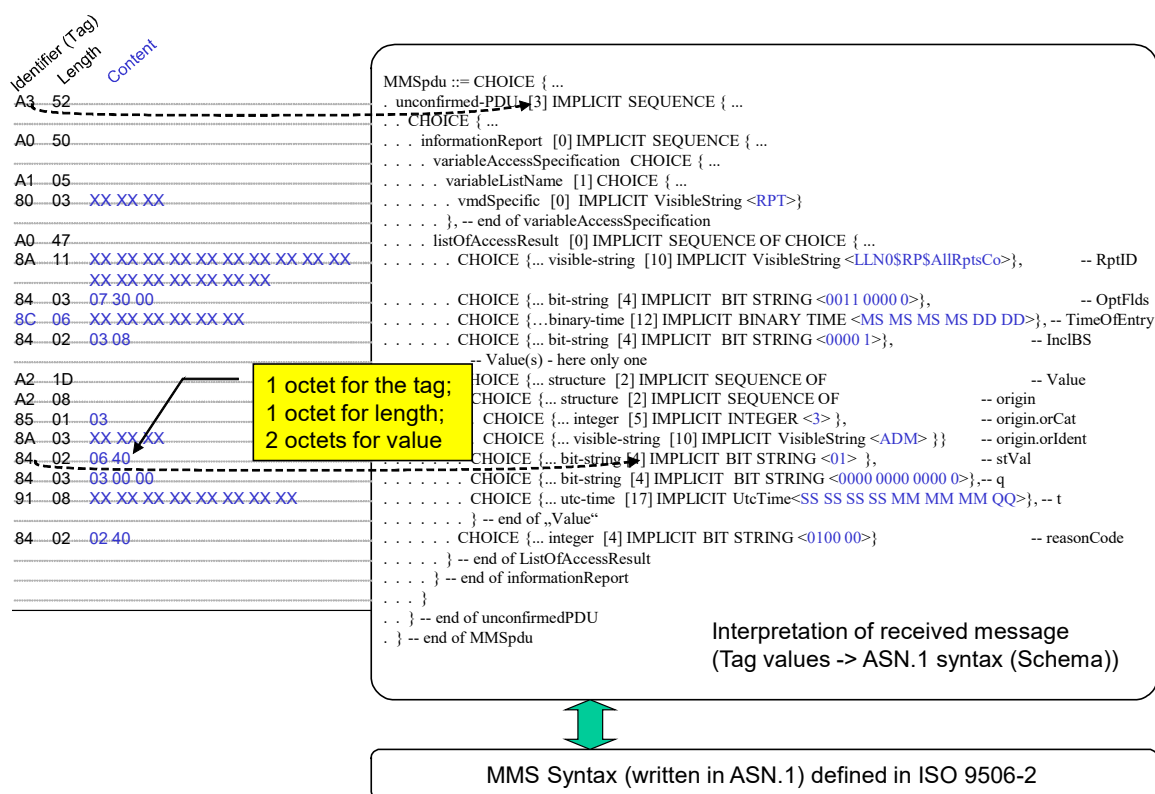


Figure F.9 – MMS information report message

These octets are packed into further messages that add lower layer-specific control and address information, for example, the TCP header and IP header.

The receiving IED is able to interpret the report message according to the identifier, lengths, names, and other values. The interpretation of the message requires the same stack, i.e., knowledge of all layers involved – including the definitions of IEC 61850-7-4, IEC 61850-7-3, IEC 61850-7-2, and IEC 61850-7-1.

NOTE 1 Implementations are expected to realise the layers in a way that hides the assembly, encoding, transmission, decoding, and interpretation of the messages. The application programs on both ends are expected to not be involved in these communication issues.

Figure F.10 illustrates a model excerpt of XCBR1 representing a real device. The complete hierarchical model may be mapped, for example, to MMS applying the SCSM according to IEC 61850-8-1. As a result, many MMS named variables have to be implemented in a real server. The services of the ACSI are mapped to MMS services.

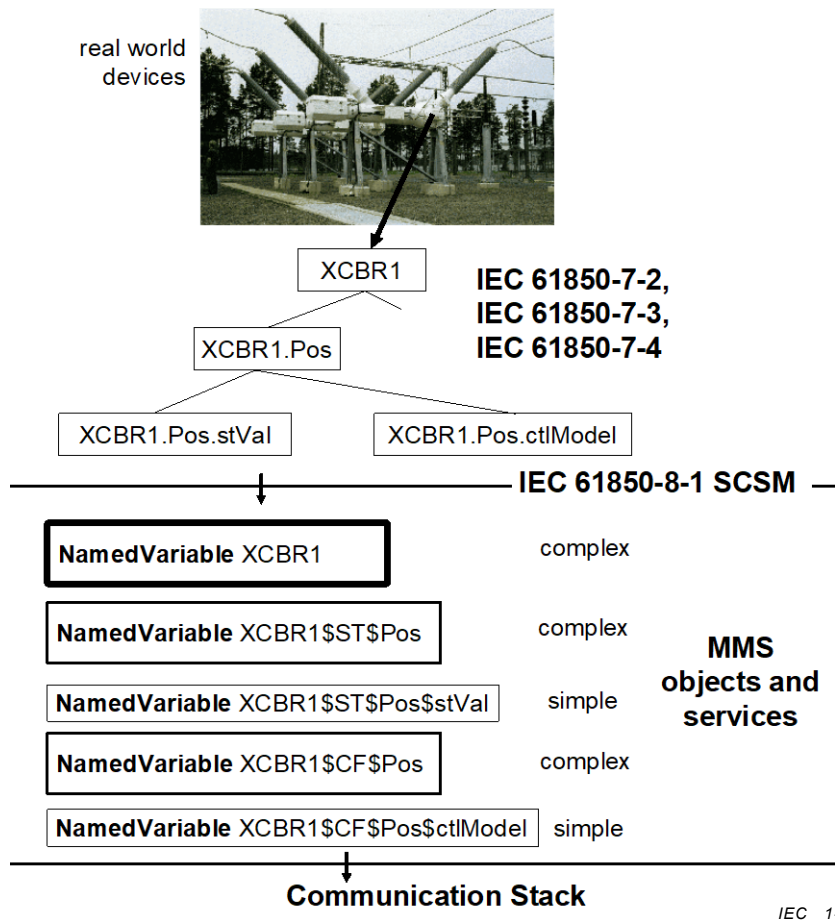


Figure F.10 – Mapping example

This example shows that the named variable XCBR1 represents the logical node (including all DATA as components of this named variable). Each component has been mapped to a less complex named variable, for example Pos (with components stVal and ctlModel). These components are mapped to two even less complex named variables: XCBR1\$ST\$Pos\$stVal and XCBR1\$CF\$Pos\$ctlModel.

NOTE 2 This multi-mapping does not require multiple storages of one value (for example, for ctlModel). The tree with all components and sub-(sub-)components is implemented once. The named variable XCBR1\$CF\$Pos\$ctlModel is just the “address” of the leaf in this tree.

MMS services support to read in one request many “full” or partial “trees”. The partial tree is described in the request message with the MMS alternate access.

Annex G (normative)

LGOS/LSVS engineering

G.1 General

This annex defines the requirements for the configuration of the logical nodes used to supervise GOOSE or SMV signals. When the supervision is statically configured, one LGOS/LSVS instance per received GOOSE/SV message type is needed.

The configuration of these logical nodes can be done either by the IED configurator tool (ICT) or the System configurator tool (SCT) or both. The configuration capabilities of the LGOS/LSVS supervision logical nodes, if supported at all, shall be declared in the ICD/IID file.

For a detailed description of these capabilities and a general description of the engineering process as well, refer to IEC 61850-6.

The diagram in Figure G.1 shows the conditions to be met in the capabilities description provided by the ICT in order that the SCT can add/delete or not LGOS/LSVS supervision logical nodes.

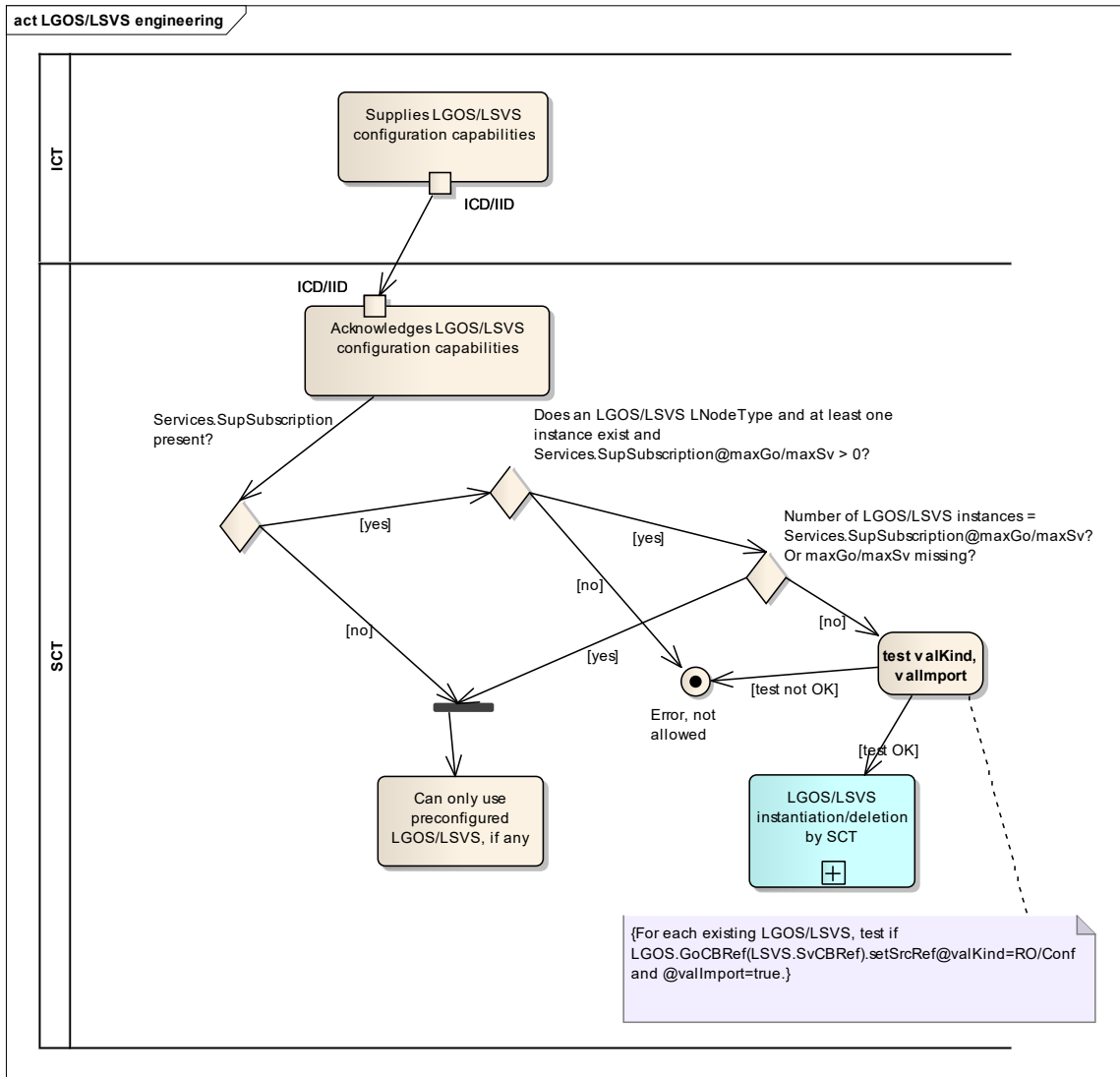


Figure G.1 – LGOS/LSVS engineering

G.2 LGOS/LSVS engineering by the ICT

The diagram in Figure G.2 shows the case where the instantiation/deletion of LGOS/LSVS logical nodes is done by the ICT.

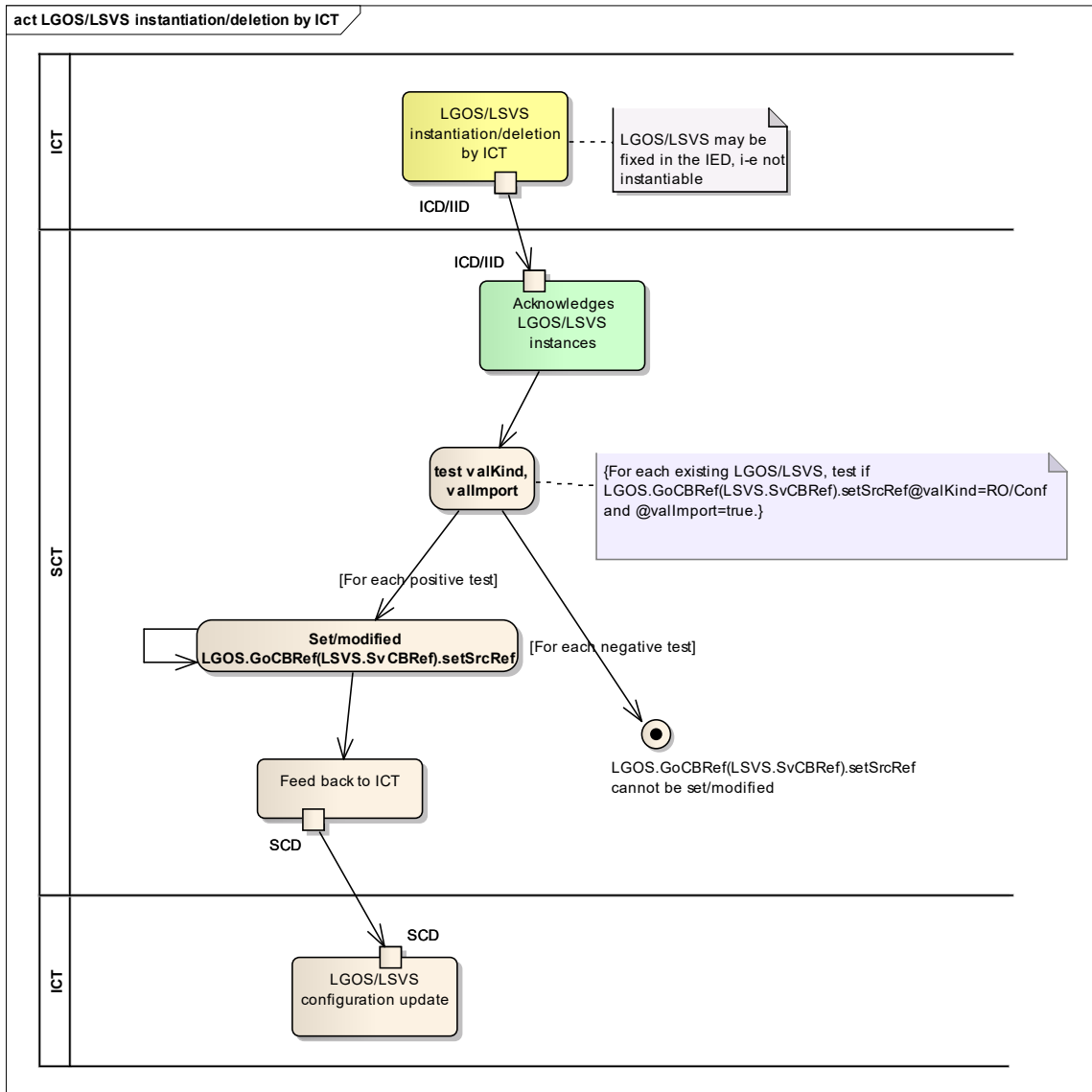


Figure G.2 – LGOS/LSVS engineering by ICT

The ICT may provide instantiation/deletion during configuration or the IED provides only fixed instances.

G.3 LGOS/LSVS engineering by the SCT

The diagram in Figure G.3 shows the activities related to adding/deleting LGOS/LSVS supervision logical nodes by the SCT when the conditions are met in the capabilities description provided by the ICT.

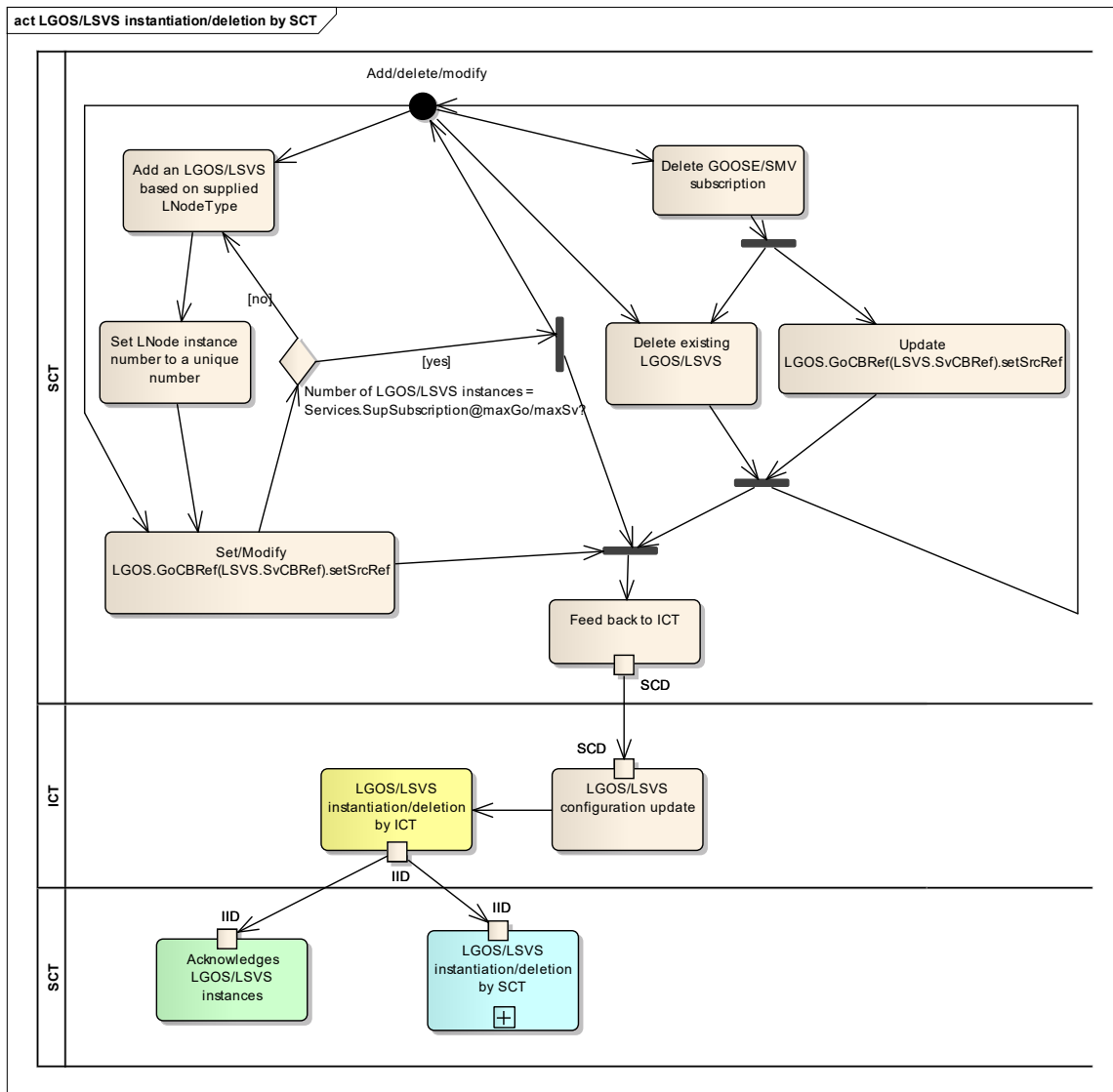


Figure G.3 – LGOS/LSVS engineering by the SCT

The workflow is as follows:

- The SCT add/delete LGOS/LSVS logical nodes as needed and the result is given to the ICT through the SCD file.
- The ICT may add/delete LGOS/LSVS logical nodes if needed and the result is given back to the SCT through the IID file.
- The SCT may either add/delete LGOS/LSVS logical nodes if needed or acknowledge the presence of the LGOS/LSVS logical nodes and set the value(s) of the source control block(s) to be monitored by a logical node(s).

The following points have to be observed:

- The engineering flow allows both the ICT and the SCT for adding/deleting supervision logical nodes instances.
- When deleting a GOOSE/SMV subscription, the ICT/SCT may either delete the corresponding LGOS/LSVS instance or update the reference to the source control block by making it empty, which means that the LGOS/LSVS instance does not supervise any telegram.
- When deleting an LGOS/LSVS instance, the ICT shall not renumber the instance number of the remaining LGOS/LSVS instances.

Annex H (normative)

GOOSE/SMV Subscription Configuration

H.1 General

The SCL language (IEC 61850-6) offers the capacity to document signals flow between IEDs within an SAS. It also allows effective engineering of which communication signal inputs are to be used by a client/subscriber to fulfil its functions. This annex describes the minimal workflow intended to be used by tools involved in the configuration of GOOSE/SMV subscription. For a detailed description of the activities involved in the workflows, refer to IEC 61850-6.

In Figure H.1, two workflows are shown depending on the capabilities that a tool declares.

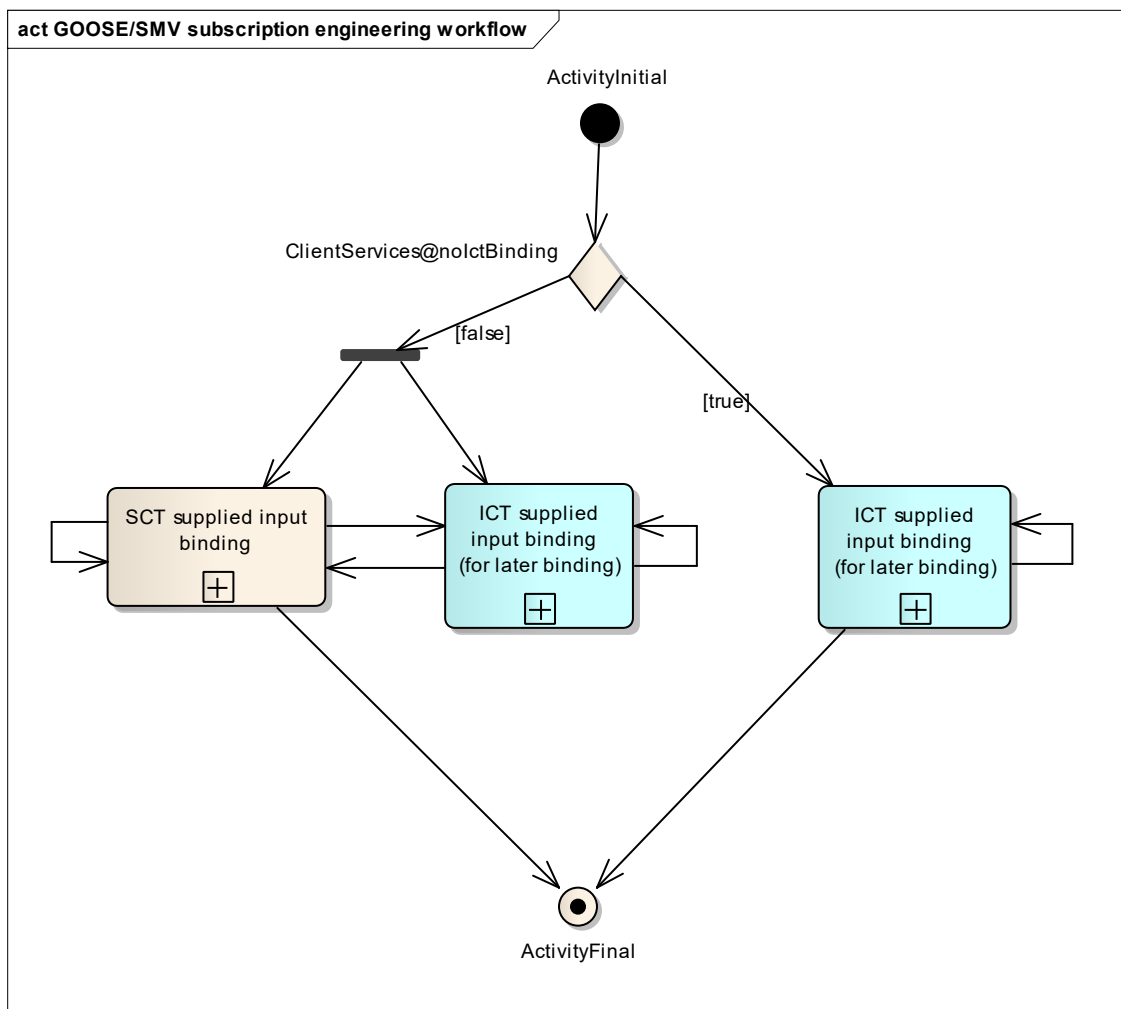


Figure H.1 – GOOSE/SMV subscription engineering workflow

If the IED tool (ICT) has no restriction and accepts that the binding of inputs be done by the system tool (SCT), the workflow “SCT supplied input binding” applies. If not, the workflow “ICT supplied input binding” shall be followed. The capability of supporting both workflows or only the “ICT supplied input binding” workflow is indicated by the attribute `noIctBinding` of the element `ClientServices` in the SCL description file of the IED.

Whatever the workflow involved, the configuration of GOOSE/SMV subscriptions shall be done by means of the SCL element GSEControl.IEDName (respectively SampledValueControl.IEDName) in the control block of the publisher. On the subscriber side, the SCL element Inputs at the LN element level shall be used to indicate which DataSet members are to be used from a subscribed message and by which logical node.

H.2 SCT supplied input binding workflow

Figure H.2 shows the intended workflow in which the ICT/IED accepts inputs binding supplied by an SCT (ClientServices@noIctBinding = FALSE). Note that the workflow does not show the configuration data transfer to the actual IED.

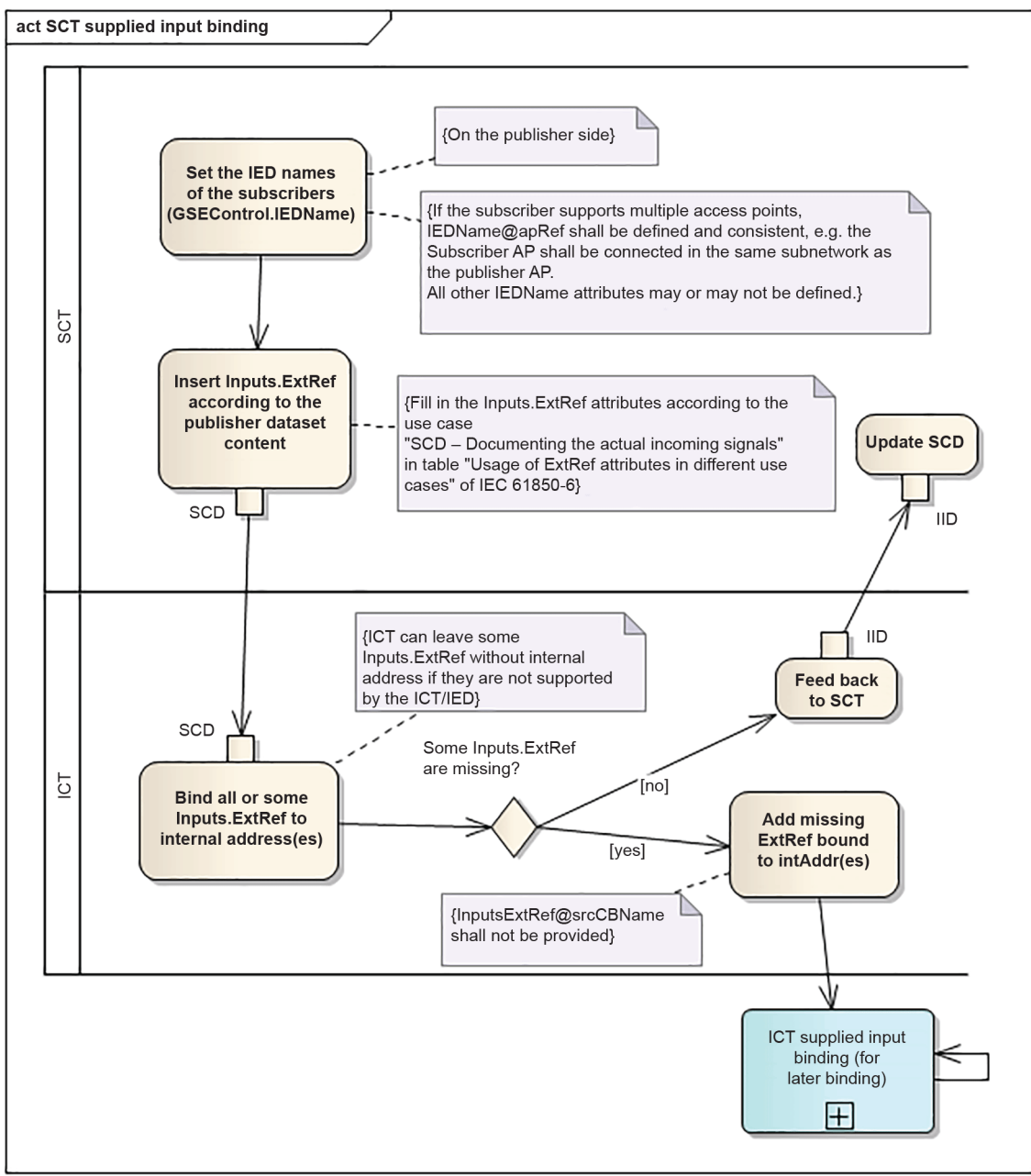


Figure H.2 – SCT supplied input binding

H.3 ICT supplied input binding (for later binding)

The workflow “ICT supplied input binding (for later binding)” is shown in Figure H3. Note that the workflow does not show the configuration data transfer to the actual IED.

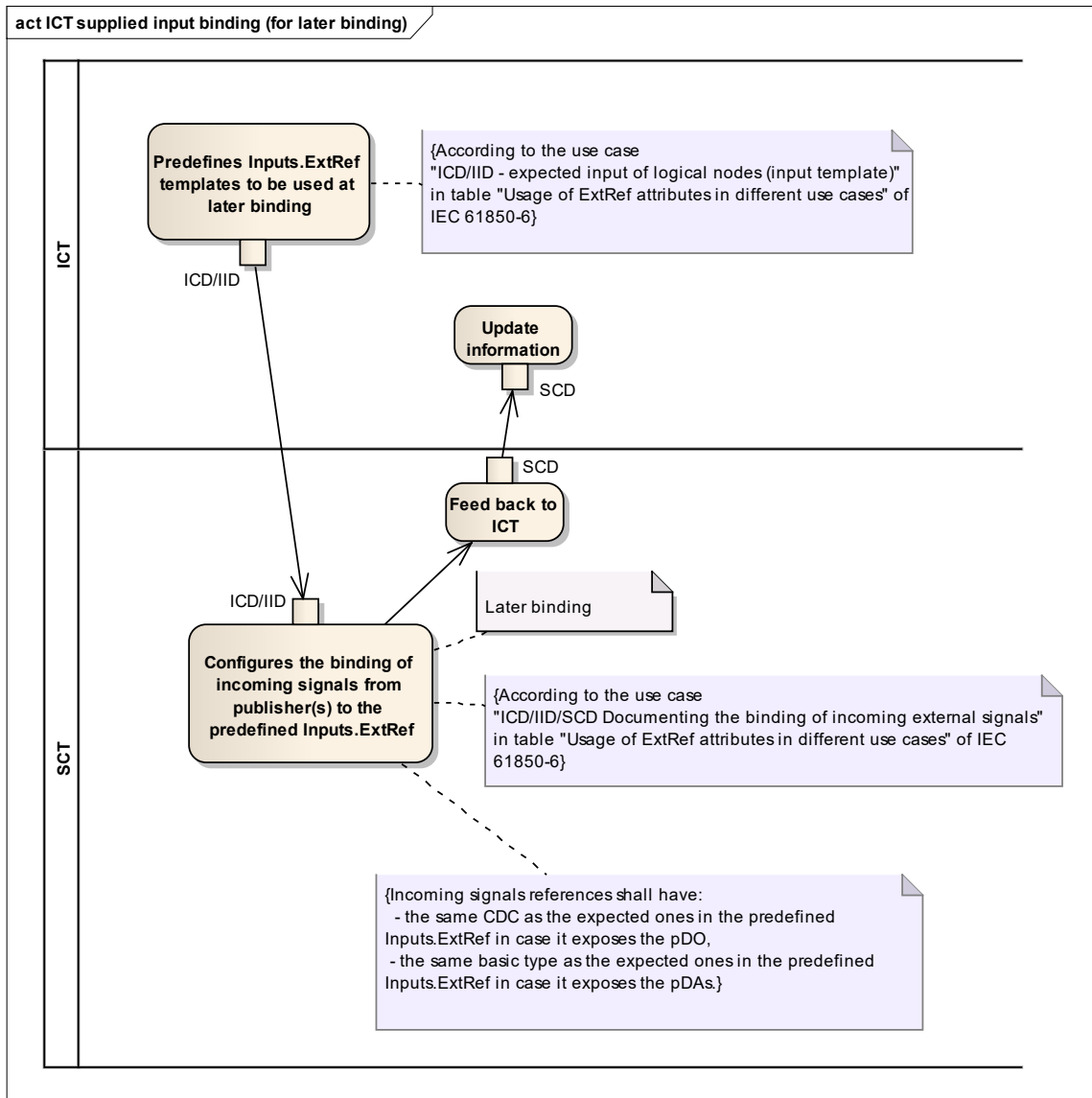


Figure H.3 – ICT supplied input binding (for later binding)

Annex I (informative)

Specification of namespaces for Edition 3

NOTE This annex is intended for Edition 3 of this part of the IEC 61850 series. In order to allow independent evolution of the domain application namespaces, it is proposed to modify the dependencies between the IEC 61850 related namespaces. The main proposal is to split the IEC 61850-7-4 name space into two namespaces. Subclause 13.3 will be adapted with this clause.

I.1 Namespaces dependencies

I.1.1 General

Three types of dependencies are defined:

- **Depends on:** means that a namespace version cannot exist without the classes that are defined in the namespace version it depends on.
- **Includes:** means that a namespace version includes some or all the classes defined in another namespace version. Applicable for domain namespaces.
- **Extends:** means that a namespace version may extend the classes defined in another namespace version.

Within the IEC, namespaces are defined by entities (e.g. IEC TC 57) responsible for vocabularies used by specific domain applications. These entities are called owners of a name space. Entities may own several namespaces.

I.1.2 Namespaces dependencies

Figure I.1 shows an example of an edition 2 logical device, however it also applies to any edition of the standard. In the example, a private LN class PRIV:

- Uses the standardized data object Str. That means the CDC type shall be ACD and of the version/revision of the basic namespace used by the device,
- Adds the new data object PrivateDO.

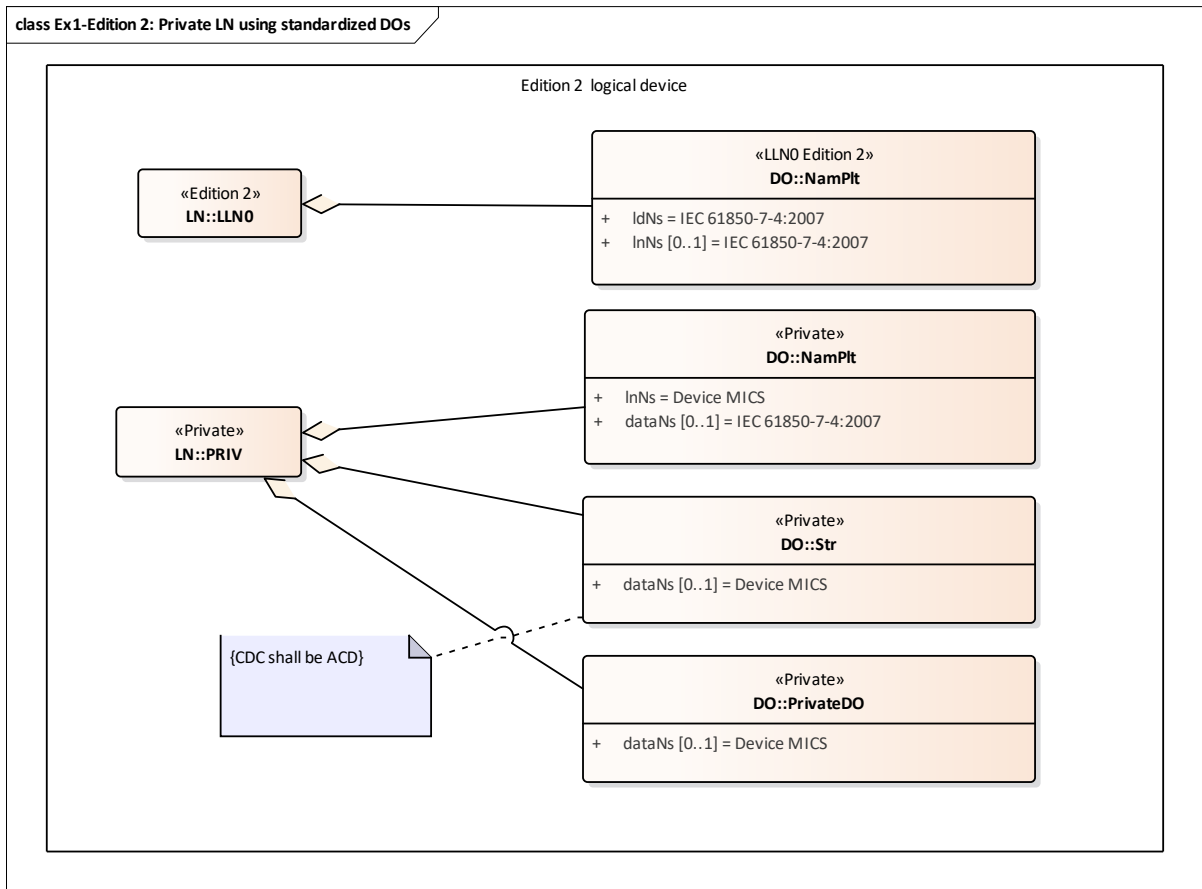


Figure I.1 – Private LN using standardized DOs (Edition 2)

I.2 Example 2 – Standardized data objects used in standardized LNs

Figure I.2 shows an example of an edition 2 logical device, however it applies to any edition of the standard. In the example, a standardized LN class GAPC uses the standardized data object OpDITmms. However, the data object namespaceID cannot be a standardized one for the data object is not defined in this namespace. It shall be a private namespaceID.

The CDC type shall be ING and of the version/revision of the basic namespace used by the device.

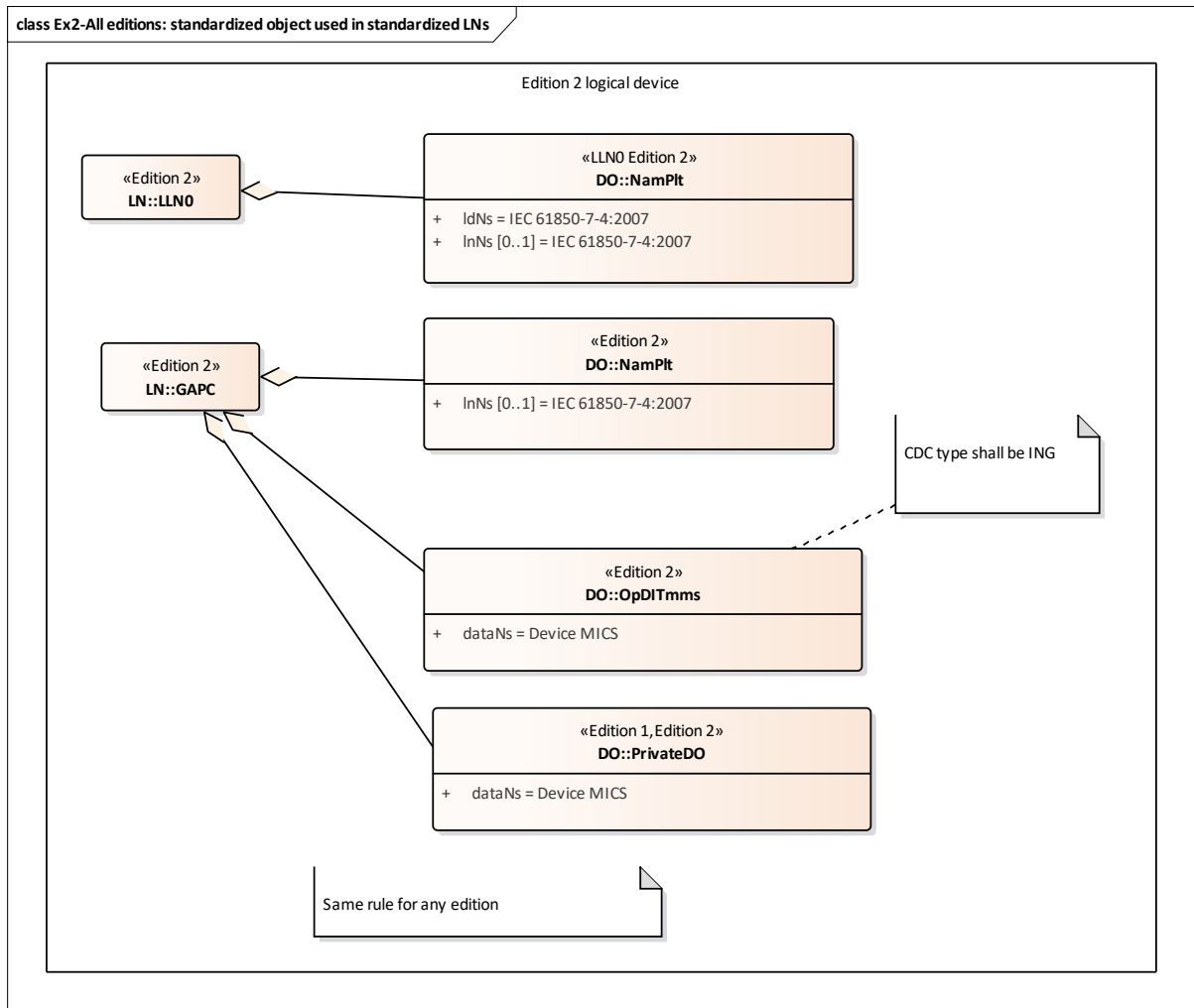


Figure I.2 –Standardized data objects used in standardized LNs

I.3 Example 3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B

The example in Figure I.3 shows an edition 2 device using, in an LLNO, a data object (SwModKey) that has been defined in a subsequent edition (here edition 2.1). There are two alternatives depending on the timeline the product is issued. If the edition 2.1 namespace does not exist yet, the edition 2 implementation shall make a private extension to the model (dataNs=Device MICS). It is important to note that in the example, the data object SwModKey may not mean “If true, switch mode is set to switching without interlocking checks (also called S1 switch mode key).”, it can be anything. If the edition 2.1 is known, and the standard data object SwModKey is needed, it can be used, and the data namespace set accordingly.

For other LN class, when edition 2.1 is known, the subsequent definition of the LN class shall be used as shown in the example with CSWI unless incompatibility issues between editions force the implementation to remain with the preceding edition.

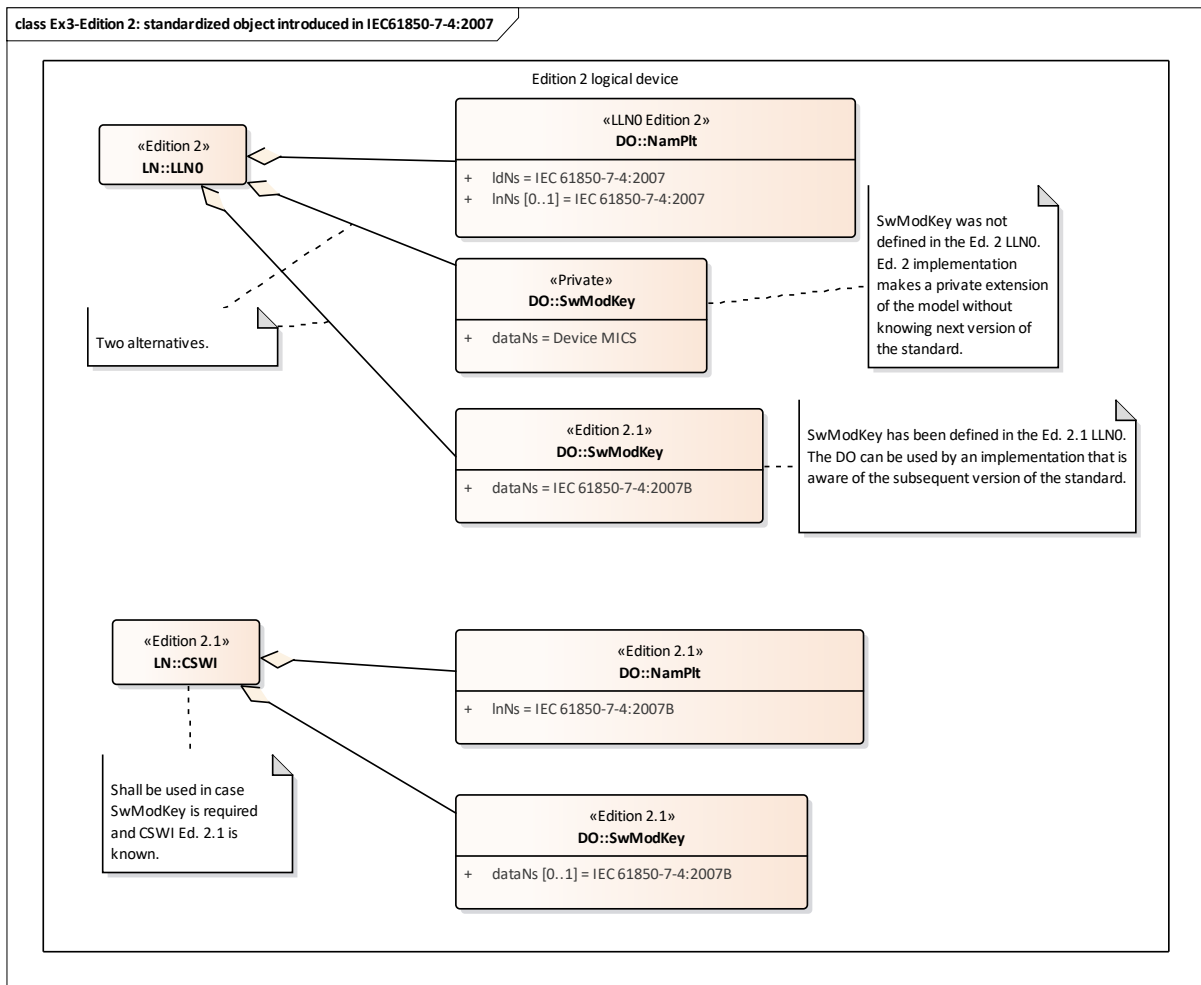


Figure I.3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B

I.4 Example 4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B

The example in Figure I.4 shows an edition 2 device using a LN class (MFLW) that has been defined in a subsequent edition (here edition 2.1). There are two alternatives depending on the timeline the product is issued. If the edition 2.1 namespace does not exist yet, the edition 2 implementation shall make a private extension to the model (InNs=Device MICS). If the edition 2.1 is known, the LN class can be used, and the LN namespace set accordingly as stated in 14.3.1 d).

The example also shows that the edition 2.1 LN class FSCH cannot be used here. The reason is that the mandatory data object NxtStrTm is based on the CDC TCS which is not part of the version/revision of the basic namespace used by the device (here IEC 61850-7-4:2007).

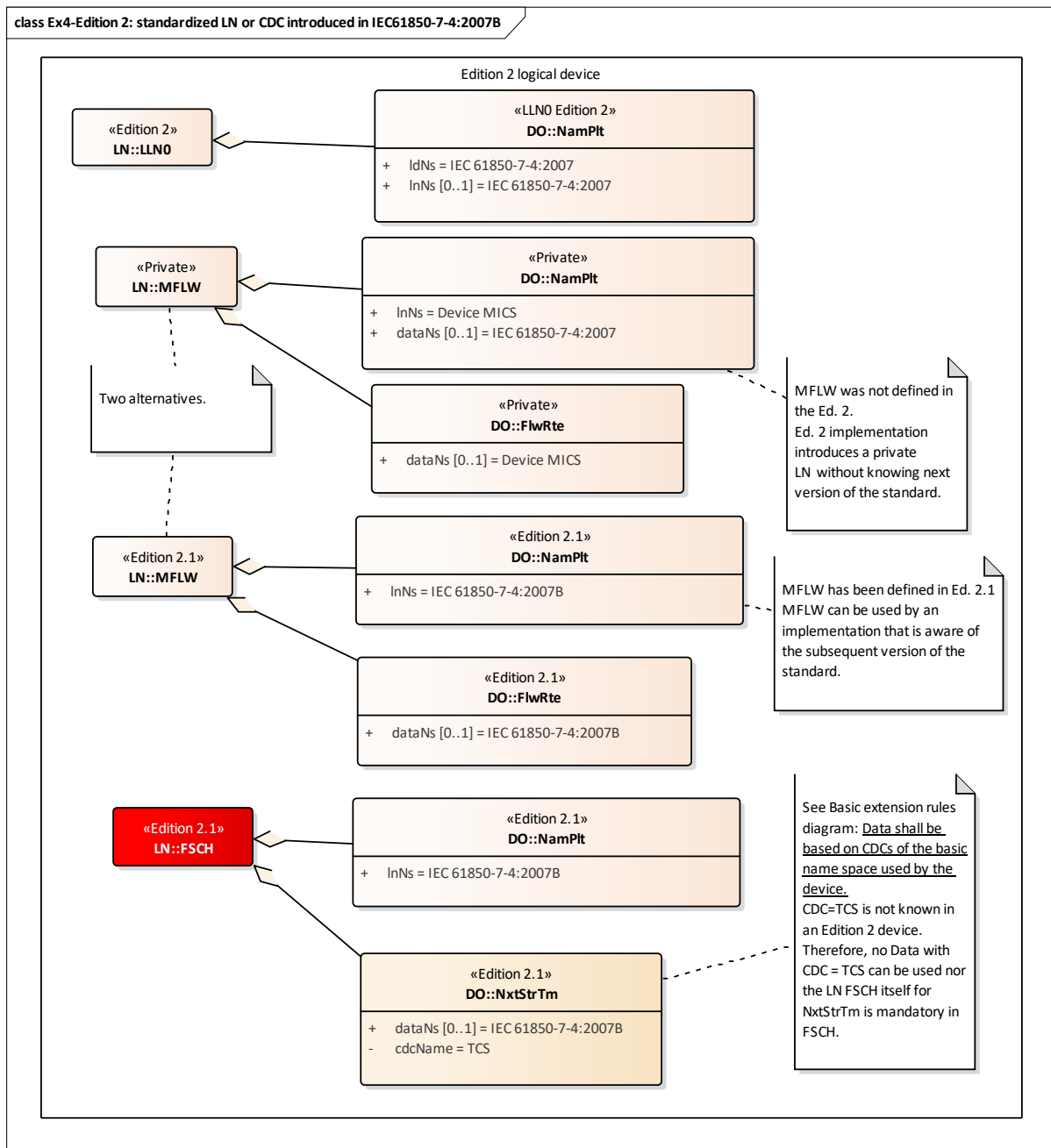


Figure I.4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B

I.5 Example 5 – Edition 1 device: Logical nodes in technical reports

Technical reports are defined as transitional namespaces. The examples in Figure I.5 and Figure I.6 expose alternatives to namespace specification when a transitional namespace is implemented in a device (here edition 1 and edition 2) and the consequences when the content of the technical report has been standardized afterwards.

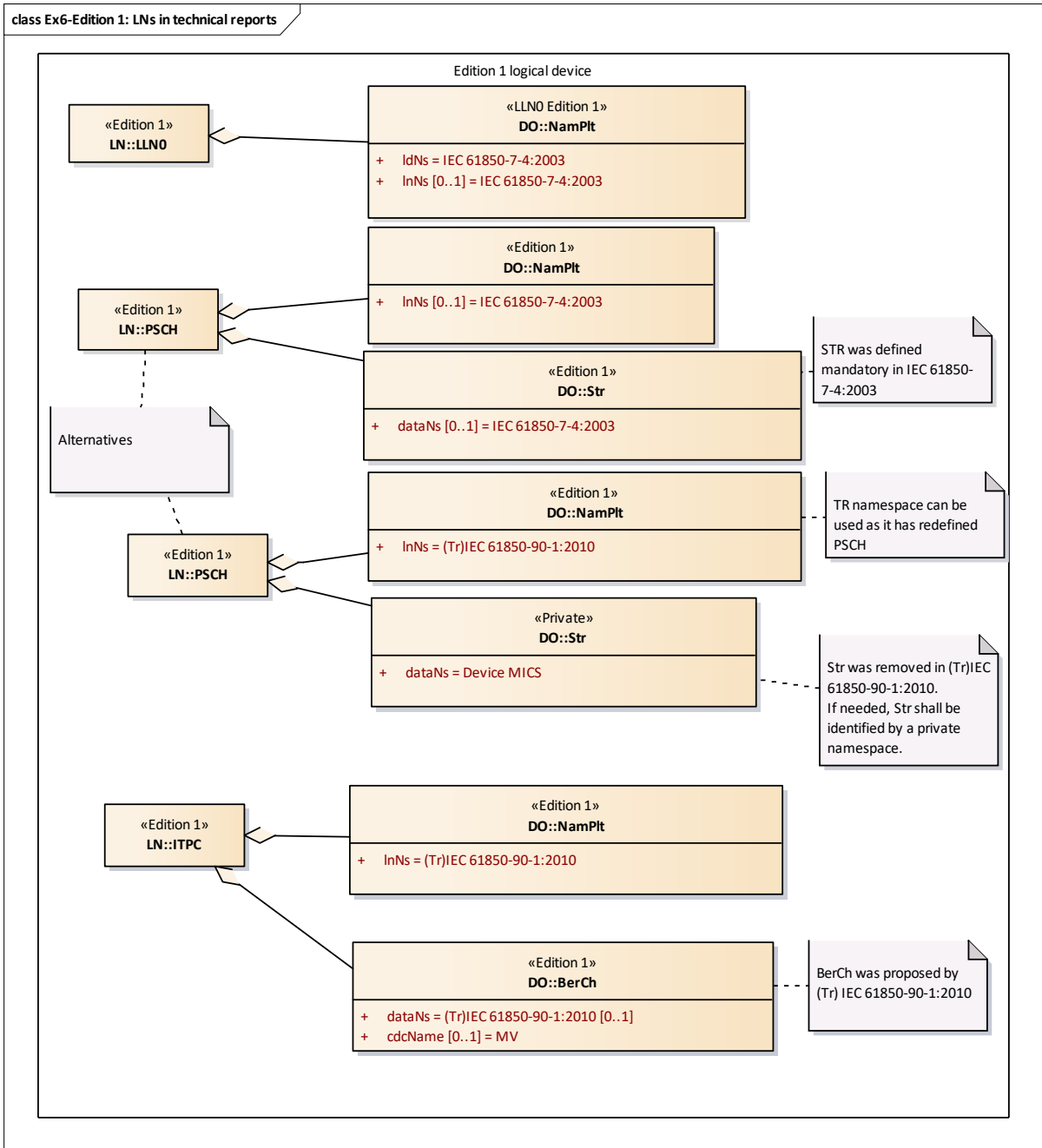


Figure I.5 – Edition 1 device: Logical nodes in technical reports

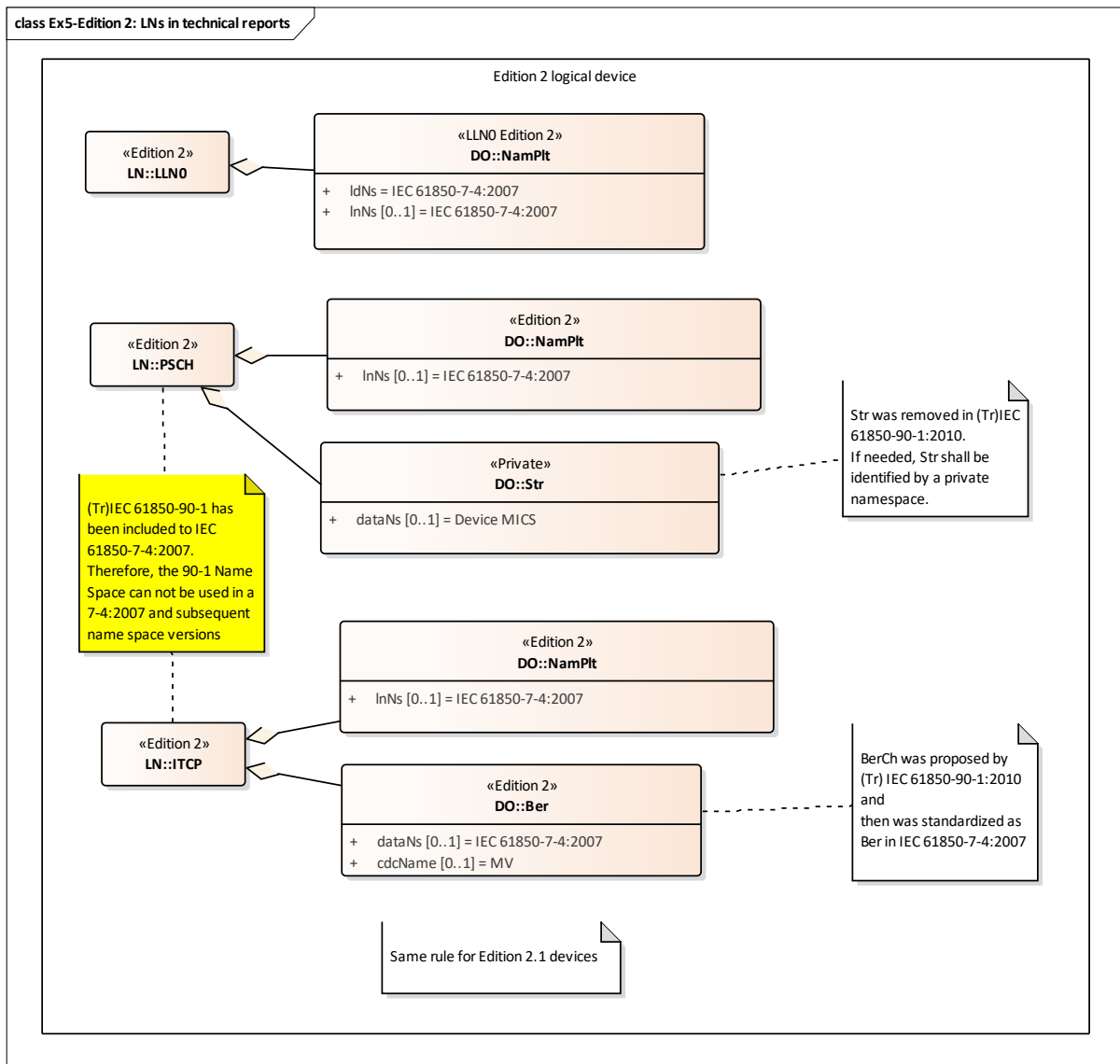


Figure I.6 – Edition 2 device: Logical nodes in technical reports

I.6 Example 6- Edition 2.1 device: Logical nodes in technical reports defining new CDCs

The example in Figure I.7 shows an edition 2.1 device implementing a hypothetical technical report proposal for a new CDC NEW that will be used by the data object NewData in the standardized InClass ZCAB.

This new CDC is defined in the technical report namespace. Therefore it shall be indicated by the data object LLN0.InNs of the logical device. As the technical report may define specific presence conditions for the NewData, it is recommended to label the ZCAB namespace with the technical report namespace.

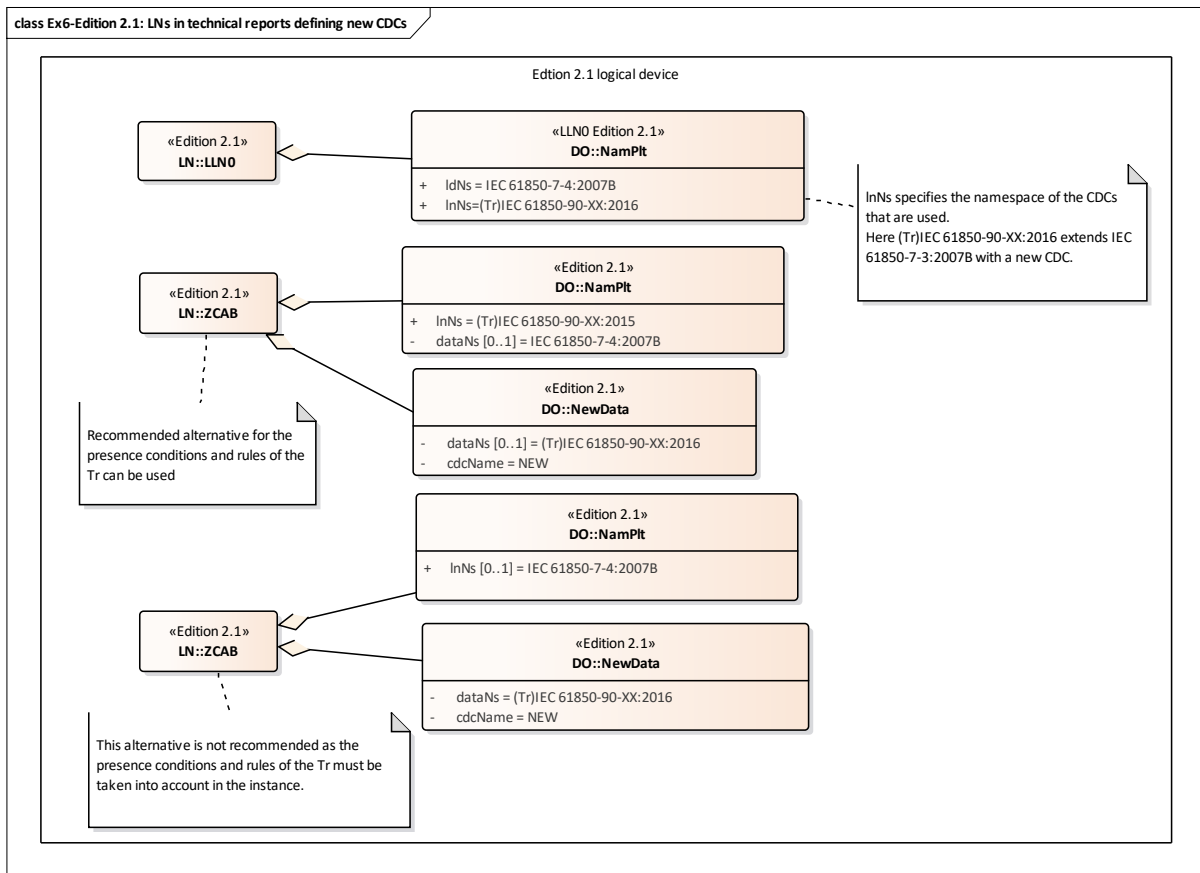


Figure I.7 – Edition 2.1 device: Logical nodes in technical reports defining new CDCs

I.7 Example 7- Edition 2.1 device: devices for product family standards

Figure I.8 illustrates a device implementing a product family standard. Product standards may extend definitions from other namespaces. IEC 61869-9: Instrument Transformers – Part 9: Digital interface for instrument transformers and IEC 62271-3: High-voltage switchgear and controlgear – Part 3: Digital interfaces based on IEC 61850 have extended respectively LPHD, TCTR, TVTR and XCBR, XSWI definitions from IEC 61850-7-4 namespace for adding equipment nameplate information. To do this, they have used the edition 2.1 CDC VSD.

The figure shows an edition 2.1 IEC 61850-7-4:2007B logical device that includes logical nodes extended by the product family standard namespace. This alternative shall be used instead of using IEC 61850-7-4:2007B logical nodes and data from the product standard namespace. The reason is that the first alternative takes into account the presence conditions and rules of the product standard namespace while the second alternative does not. Moreover, the alternative shall extend the LPHD logical node with the data object NamPIt in order to specify the InNs of LPHD. This namespace is different from the logical device namespace (IdNs).

NOTE For a definition of the presence conditions, refer to IEC 61850-7-4.

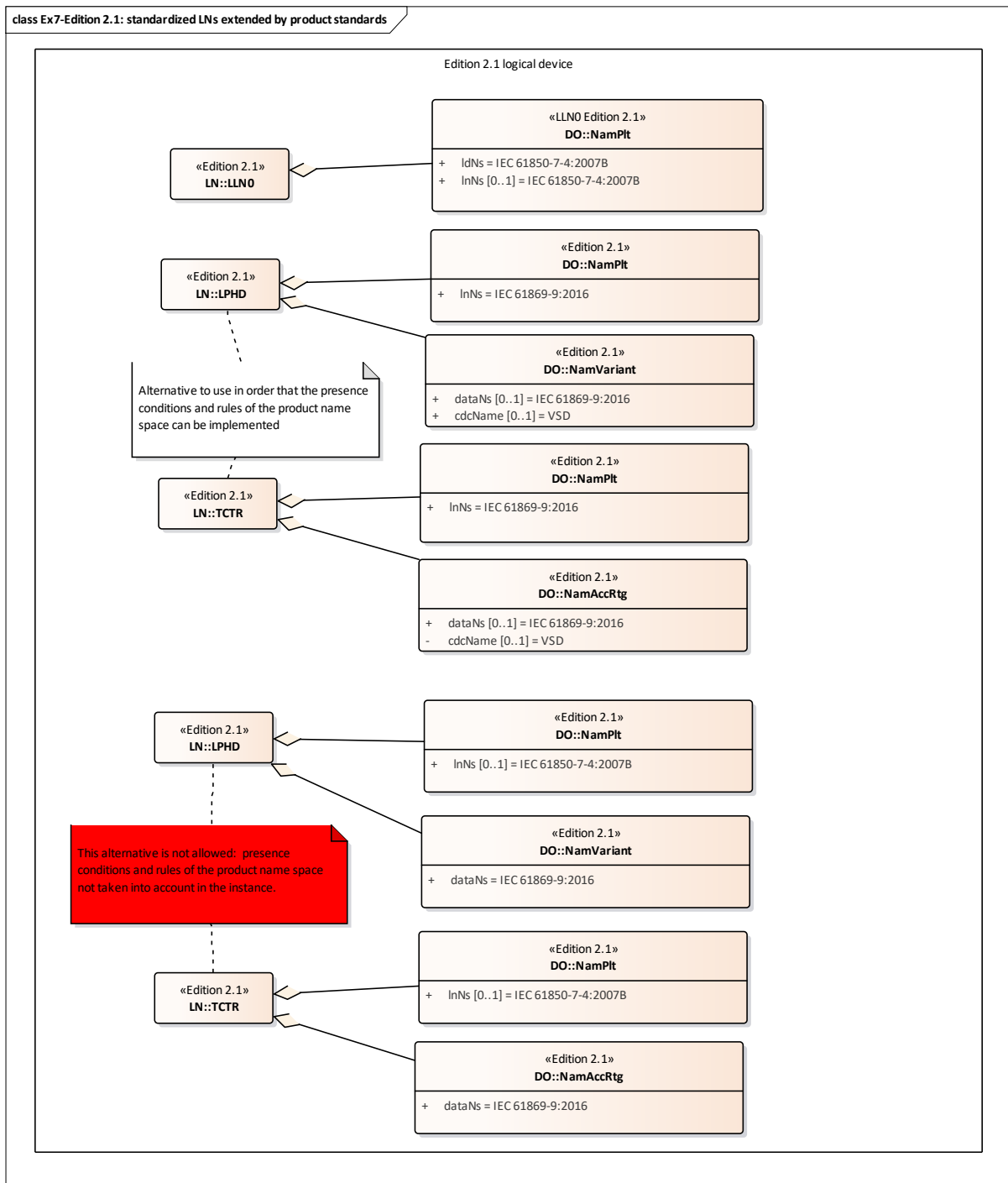


Figure I.8 – Edition 2.1 device: standardized LNs extended by other domains

I.8 Example 8 – Standardized logical nodes introduced by other domain applications

This clause gives examples about the use of logical nodes that have been introduced by the DER domain application. The DER domain application reuses logical node definitions found in the IEC 61850-7-4 namespace. As such, the IEC 61850-7-420 specification shall declare that it includes the IEC 61850-7-4 namespace. Figure I.9 and Figure I.10 show different versions of IEC 61850-7-420 devices and the related mandatory/optional namespace declarations.

NOTE LLNO.NamPit.InNs specifies the basic namespace the device is based on.

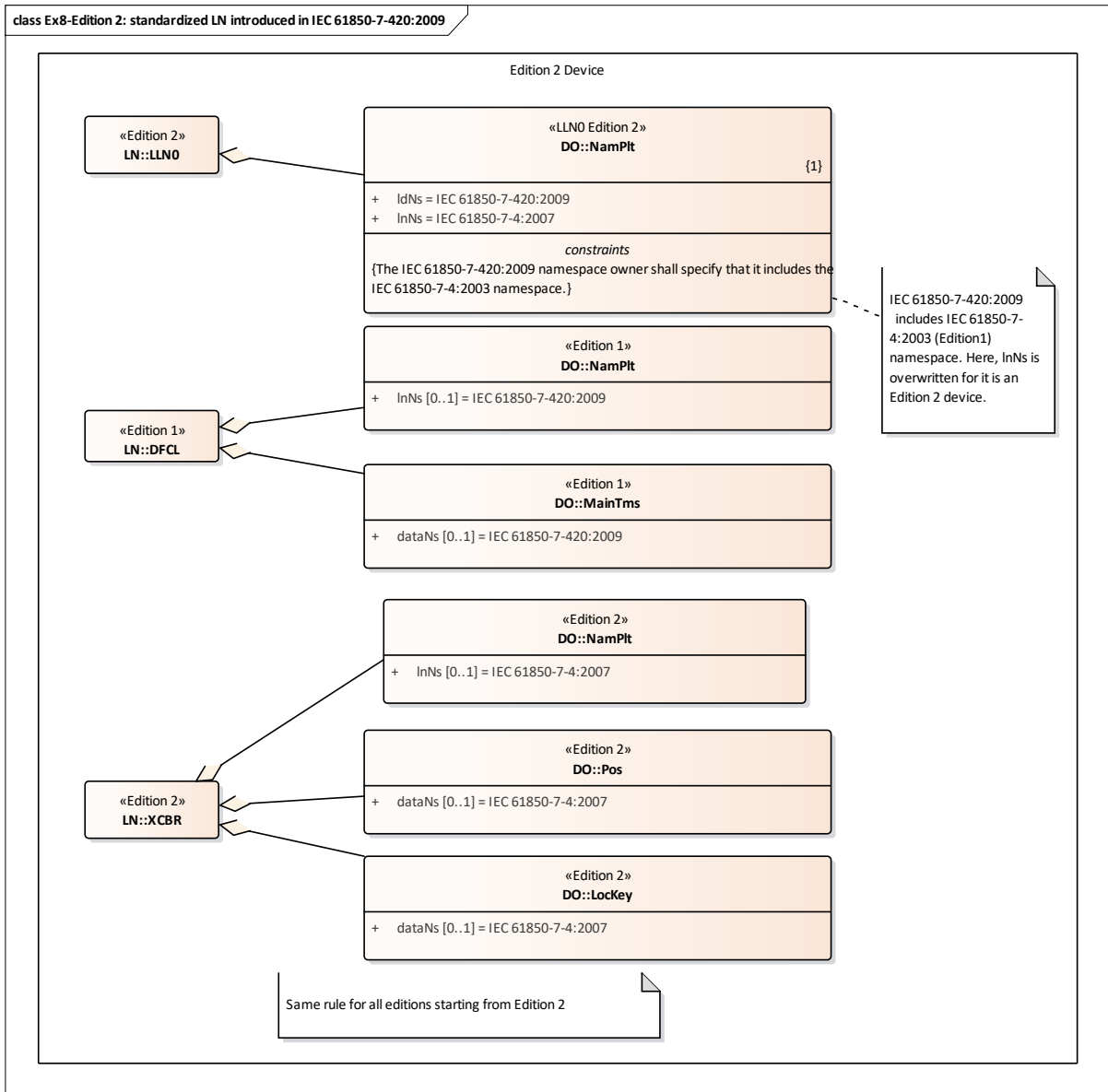


Figure I.9 – Edition 2 device: standardized LN introduced in IEC 61850-7-420:2009

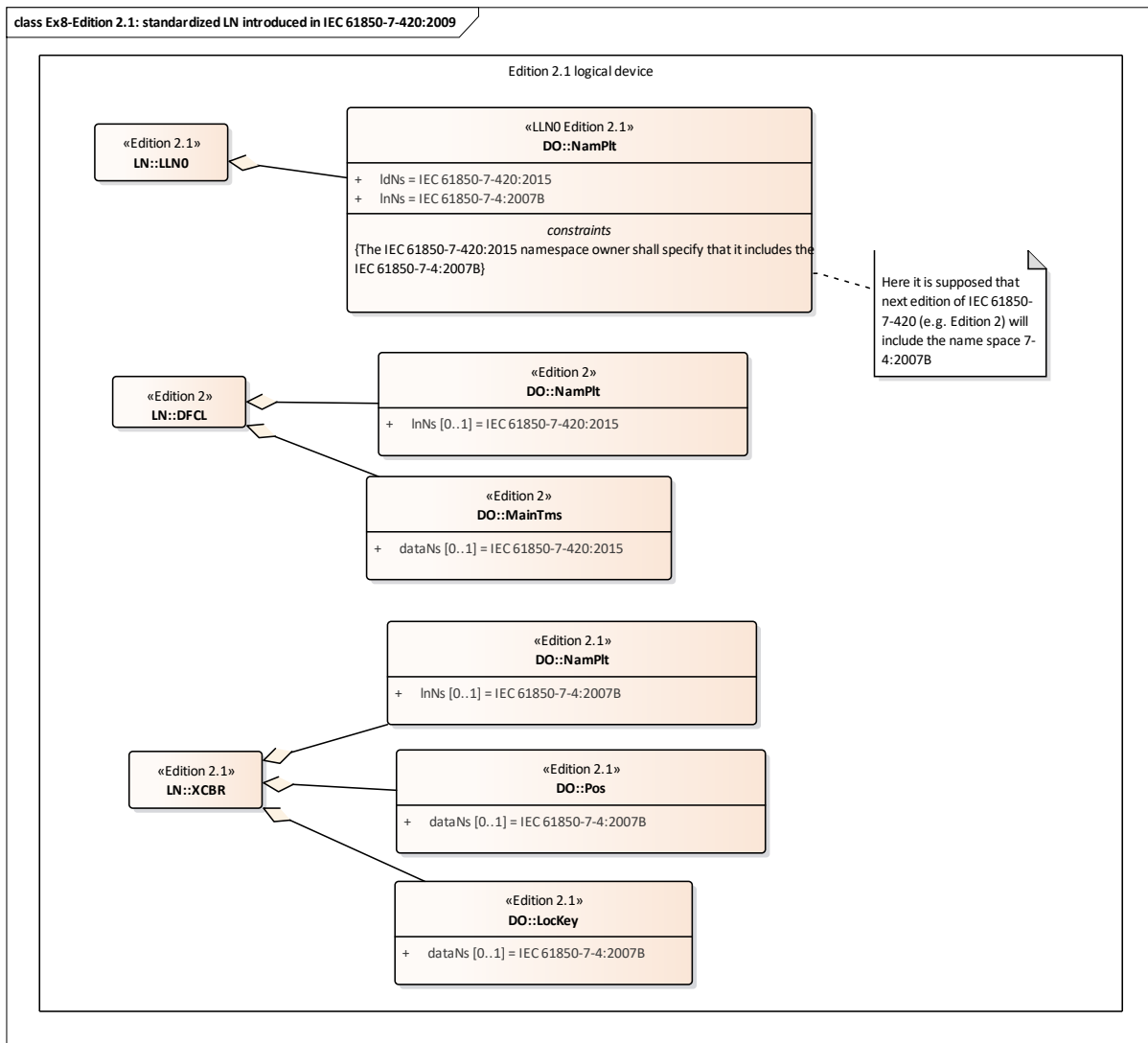


Figure I.10 – Edition 2.1 device: standardized LN introduced in IEC 61850-7-420:2015

NOTE The version for the second edition of the IEC 61850-7-420 is unknown at this time, and the 2015 is used as an example.

I.9 Example 9 – Standardized logical nodes introduced by other domain applications than moved to IEC 61850-7-4

This clause illustrates the situation where logical nodes first defined in a domain namespace (here IEC 61850-7-420) are then moved to IEC 61850-7-4. This situation is not privileged and should be covered by means of technical reports. However it cannot be completely avoided. The situation described is the following:

- The LN STMP has first been defined in IEC 61850-7-420:2009 than moved to IEC 61850-7-4:2007.
- IEC 61850-7-420:2009 defines a data object TmpRteSt (Temperature rate alarm) while IEC 61850-7-4:2007 does not.
- IEC 61850-7-4:2007B defines a data object for temperature rate alarm and have harmonized the name to RteAlm.

The examples given in Figure I.11 and Figure I.12 show different namespace versions of devices and the rules about namespace declarations.

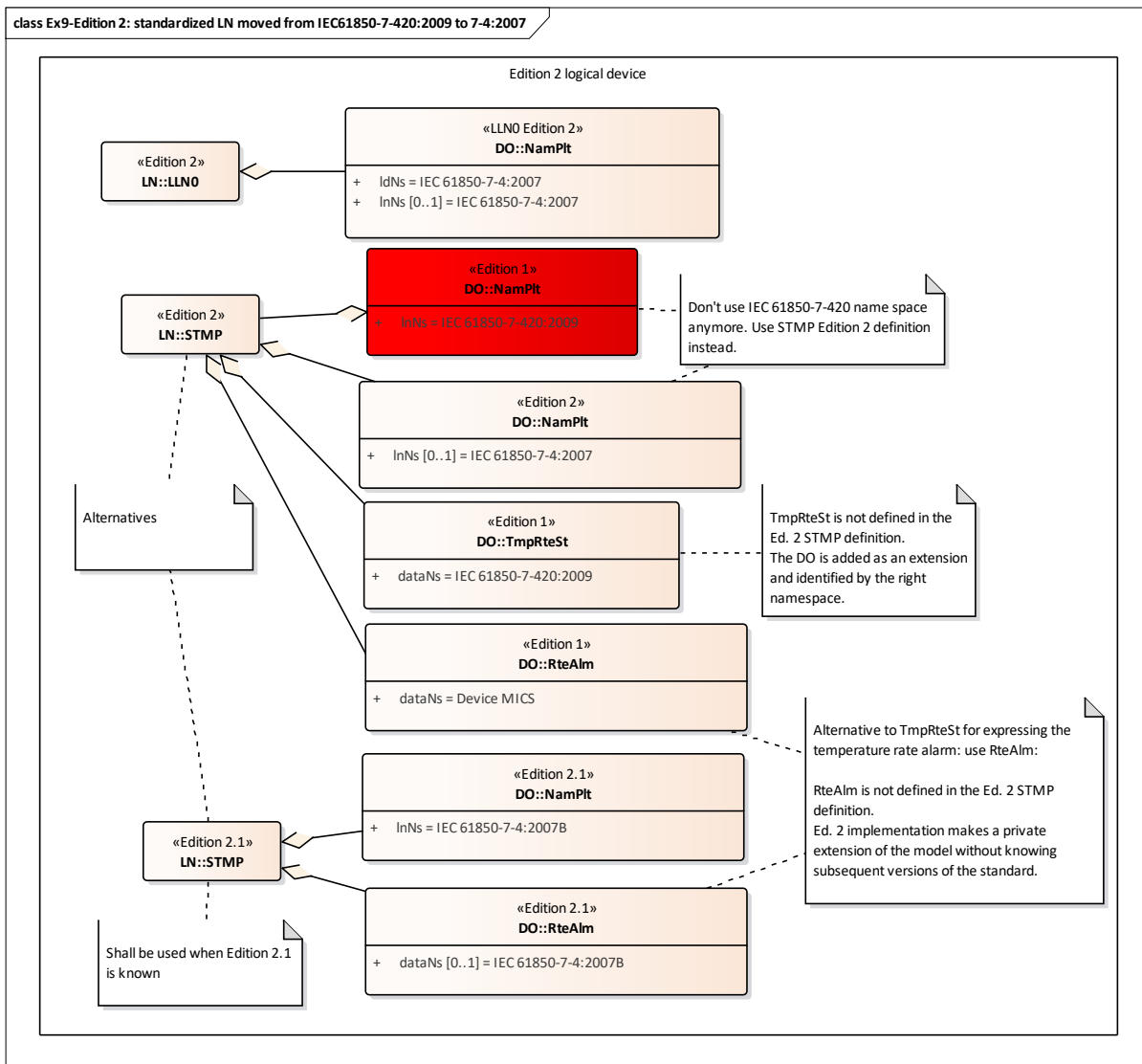


Figure I.11 – Edition 2 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007

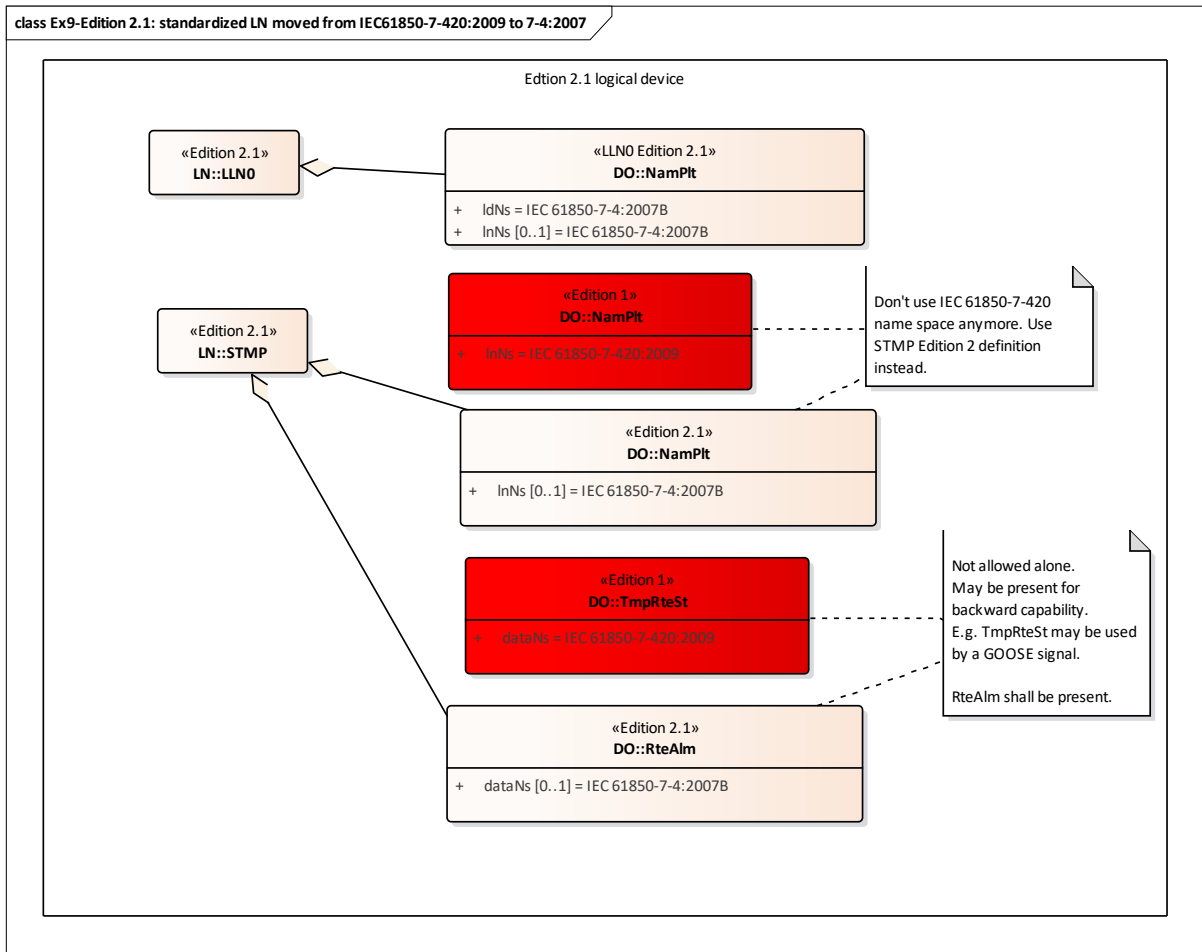


Figure I.12 – Edition 2.1 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007

Annex J (normative)

Use case scenarios examples for clarifying the common rules of Clause 14

J.1 General

This annex contains class diagrams to illustrate the mandatory/optional use of the different namespace attributes defined in Subclause 13.5. The following definitions are used.

- A multiplicity [0..1] means optional, otherwise mandatory,
- Device MICS means that the namespaceID shall be a private namespace defined in the MICS.

J.2 Example 1 – Private LN using standardized DOs

Figure J.1 shows an example of an edition 2 logical device, however it also applies to any edition of the standard. In the example, a private LN class PRIV:

- Uses the standardized data object Str. That means the CDC type shall be ACD and of the version/revision of the basic namespace used by the device,
- Adds the new data object PrivateDO.

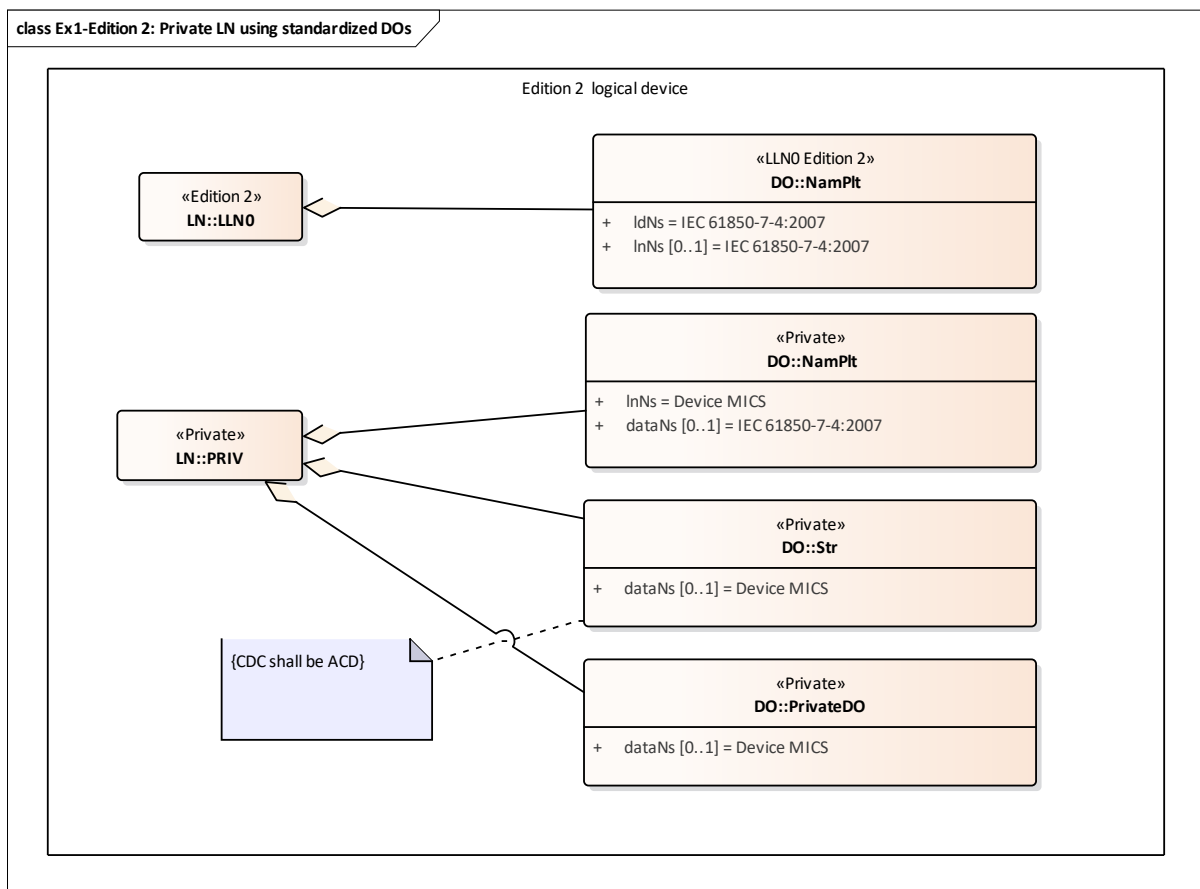


Figure J.1 – Private LN using standardized DOs (Edition 2)

J.3 Example 2 – Standardized data objects used in standardized LNs

Figure J.2 shows an example of an edition 2 logical device, however it applies to any edition of the standard. In the example, a standardized LN class GAPC uses the standardized data object OpDITmms. However, the data object namespaceID cannot be a standardized one for the data object is not defined in this namespace. It shall be a private namespaceID.

The CDC type shall be ING and of the version/revision of the basic namespace used by the device.

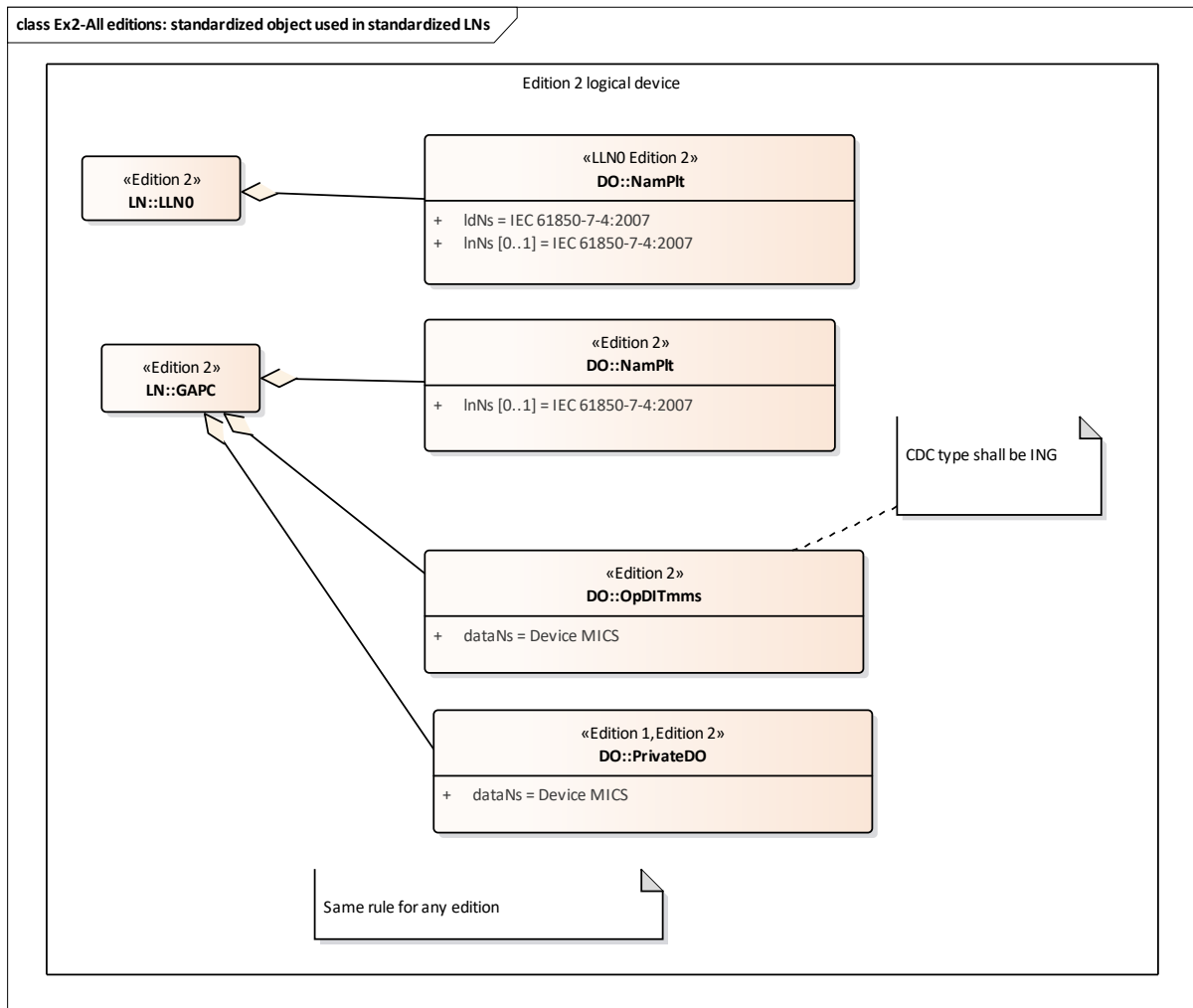


Figure J.2 –Standardized data objects used in standardized LNs

J.4 Example 3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B

The example in Figure J.3 shows an edition 2 device using, in an LLN0, a data object (SwModKey) that has been defined in a subsequent edition (here edition 2.1). There are two alternatives depending on the timeline the product is issued. If the edition 2.1 namespace does not exist yet, the edition 2 implementation shall make a private extension to the model (dataNs=Device MICS). It is important to note that in the example, the data object SwModKey may not mean “If true, switch mode is set to switching without interlocking checks (also called S1 switch mode key).”, it can be anything. If the edition 2.1 is known, and the standard data object SwModKey is needed, it can be used, and the data namespace set accordingly.

For other LN class, when edition 2.1 is known, the subsequent definition of the LN class shall be used as shown in the example with CSWI unless incompatibility issues between editions force the implementation to remain with the preceding edition.

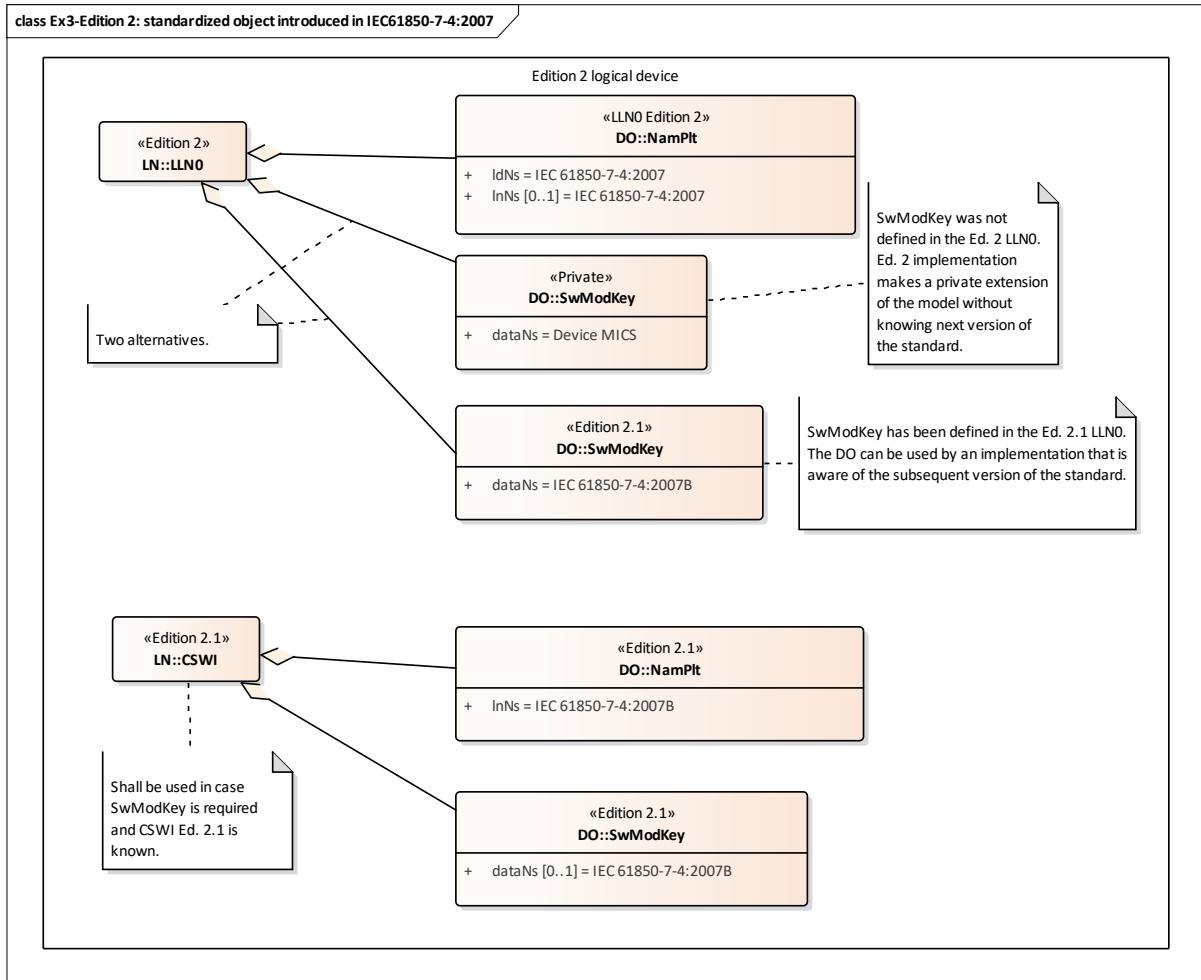


Figure J.3 – Edition 2 device: standardized data object introduced in IEC61850-7-4:2007B

J.5 Example 4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B

The example in Figure J.4 shows an edition 2 device using a LN class (MFLW) that has been defined in a subsequent edition (here edition 2.1). There are two alternatives depending on the timeline the product is issued. If the edition 2.1 namespace does not exist yet, the edition 2 implementation shall make a private extension to the model (InNs=Device MICS). If the edition 2.1 is known, the LN class can be used, and the LN namespace set accordingly as stated in 14.3.1 d).

The example also shows that the edition 2.1 LN class FSCH cannot be used here. The reason is that the mandatory data object NxtStrTm is based on the CDC TCS which is not part of the version/revision of the basic namespace used by the device (here IEC 61850-7-4:2007).

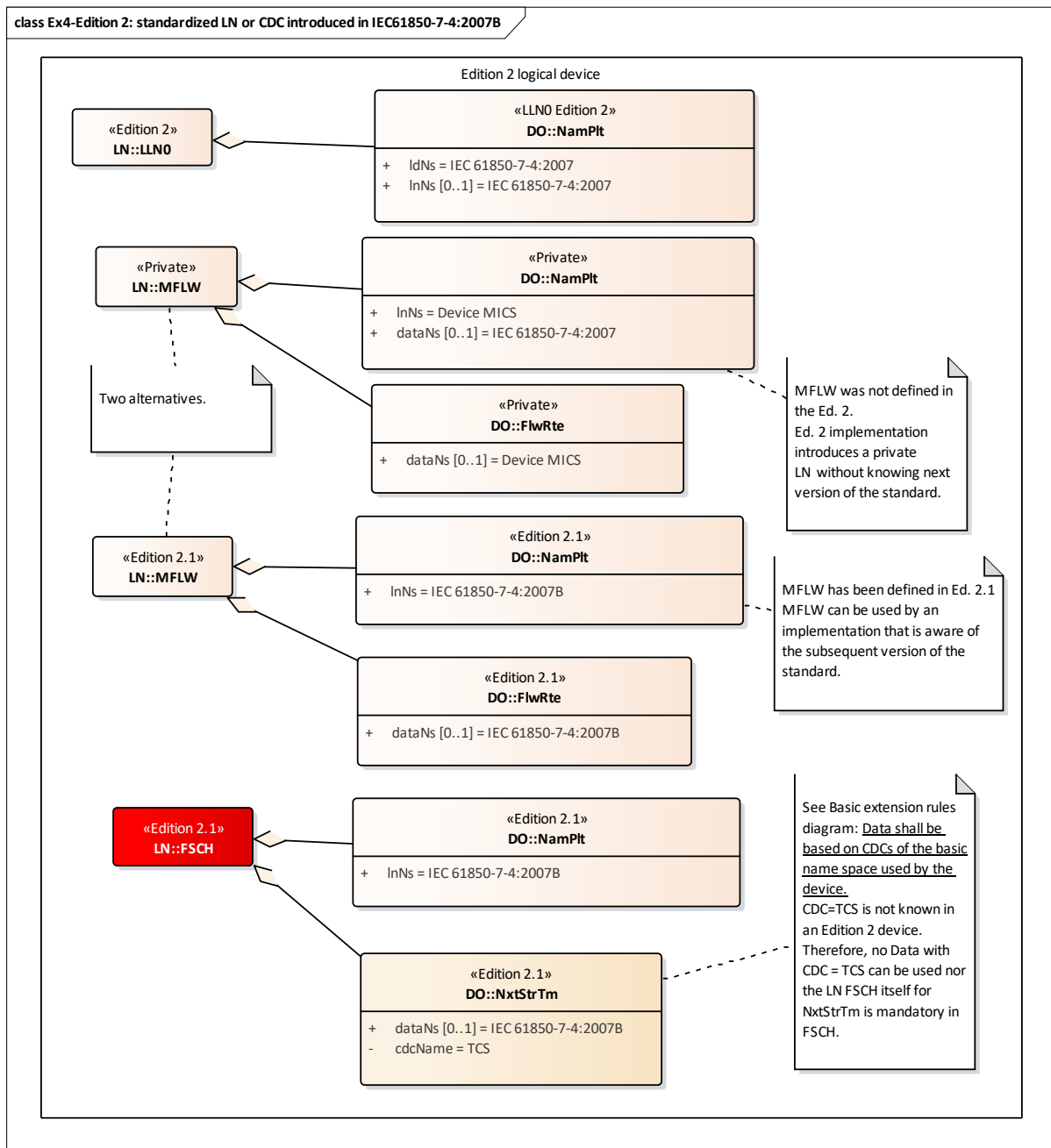


Figure J.4 – Edition 2 device: standardized LN introduced in IEC61850-7-4:2007B

J.6 Example 5 – Edition 1 or 2 device: Logical nodes in technical reports

Technical reports are defined as transitional namespaces. The examples in Figure J.5 and Figure J.6 expose alternatives to namespace specification when a transitional namespace is implemented in a device (here edition 1 and edition 2) and the consequences when the content of the technical report has been standardized afterwards.

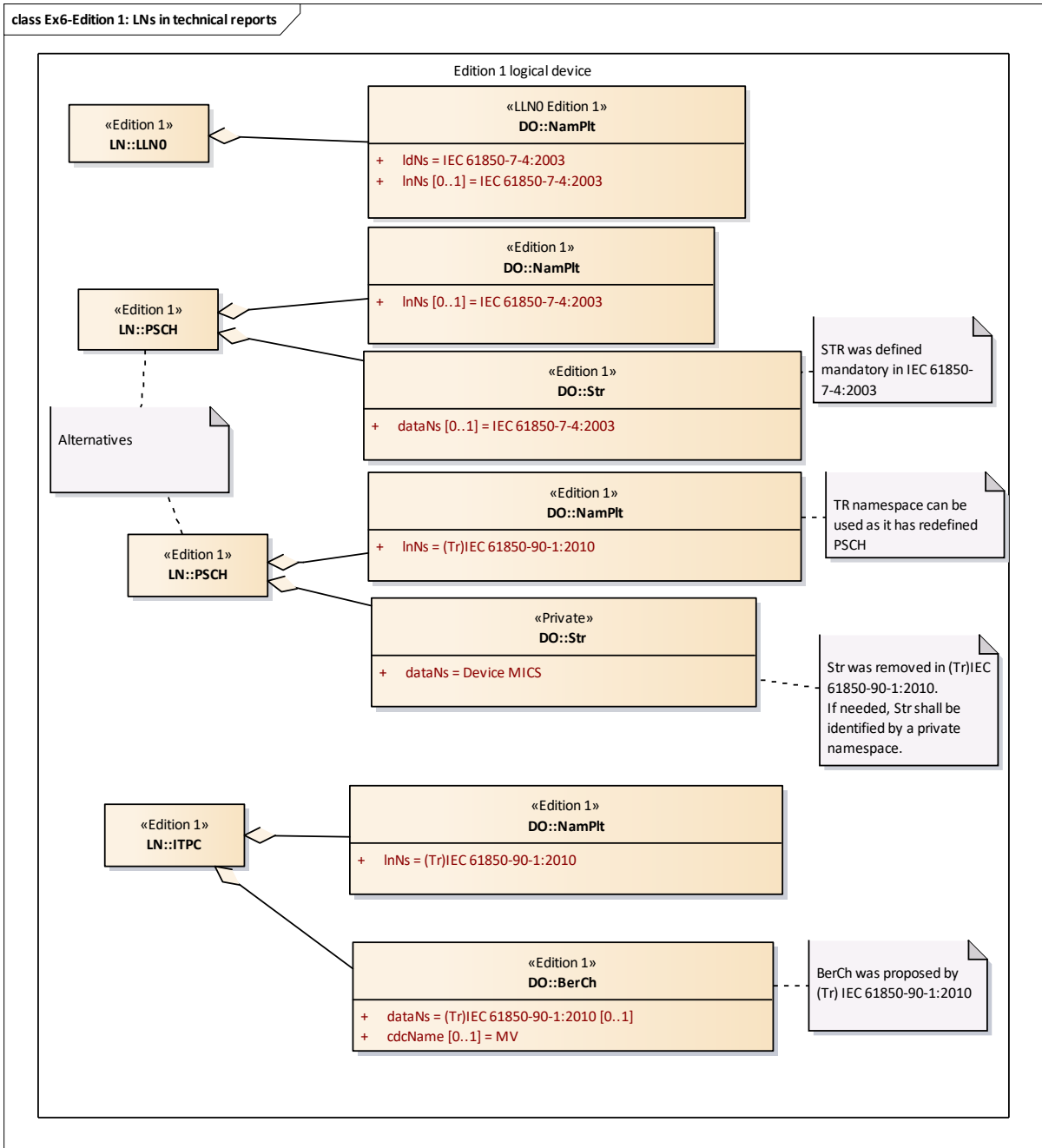


Figure J.5 – Edition 1 device: Logical nodes in technical reports

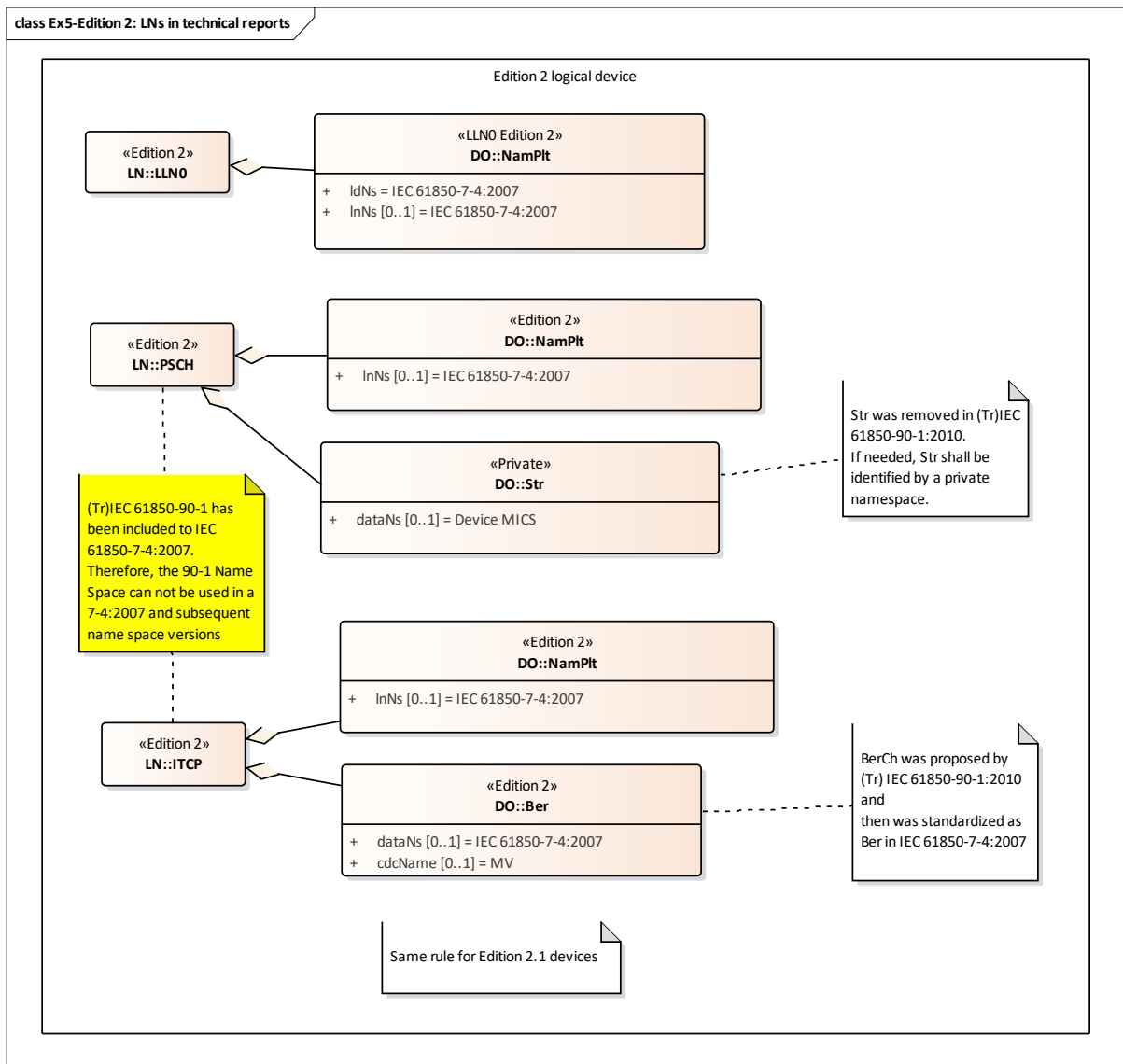


Figure J.6 – Edition 2 device: Logical nodes in technical reports

J.7 Example 6- Edition 2.1 device: Logical nodes in technical reports defining new CDCs

The example in Figure J.7 shows an edition 2.1 device implementing a hypothetical technical report proposal for a new CDC NEW that will be used by the data object NewData in the standardized InClass ZCAB.

This new CDC is defined in the technical report namespace. Therefore it shall be indicated by the data object LLN0.InNs of the logical device. As the technical report may define specific presence conditions for the NewData, it is recommended to label the ZCAB namespace with the technical report namespace.

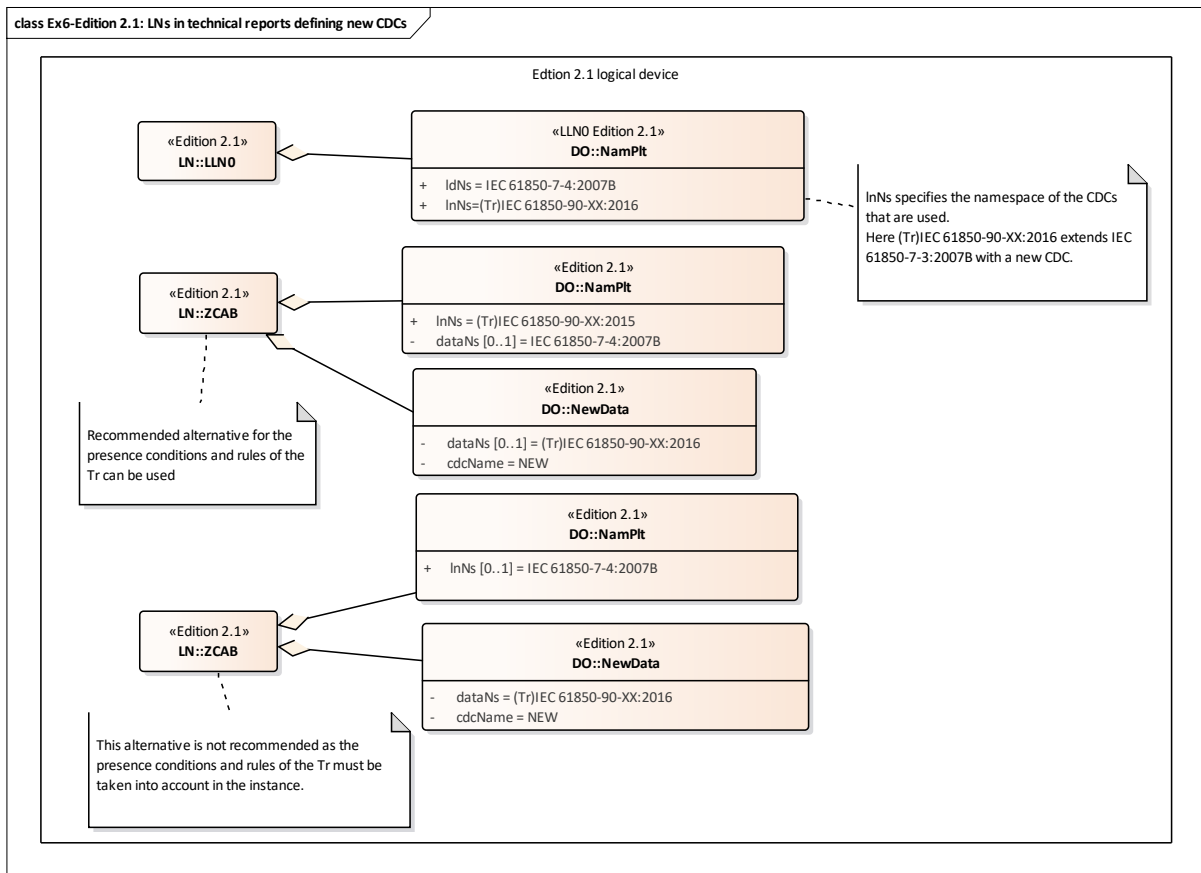


Figure J.7 – Edition 2.1 device: Logical nodes in technical reports defining new CDCs

J.8 Example 7- Edition 2.1 device: devices for product family standards

Figure J.8 illustrates a device implementing a product family standard. Product standards may extend definitions from other namespaces. IEC 61869-9: Instrument Transformers – Part 9: Digital interface for instrument transformers and IEC 62271-3: High-voltage switchgear and controlgear – Part 3: Digital interfaces based on IEC 61850 have extended respectively LPHD, TCTR, TVTR and XCBR, XSWI definitions from IEC 61850-7-4 namespace for adding equipment nameplate information. To do this, they have used the edition 2.1 CDC VSD.

The figure shows an edition 2.1 IEC 61850-7-4:2007B logical device that includes logical nodes extended by the product family standard namespace. This alternative shall be used instead of using IEC 61850-7-4:2007B logical nodes and data from the product standard namespace. The reason is that the first alternative takes into account the presence conditions and rules of the product standard namespace while the second alternative does not. Moreover, the alternative shall extend the LPHD logical node with the data object NamPlt in order to specify the InNs of LPHD. This namespace is different from the logical device namespace (IdNs).

NOTE For a definition of the presence conditions, refer to IEC 61850-7-4.

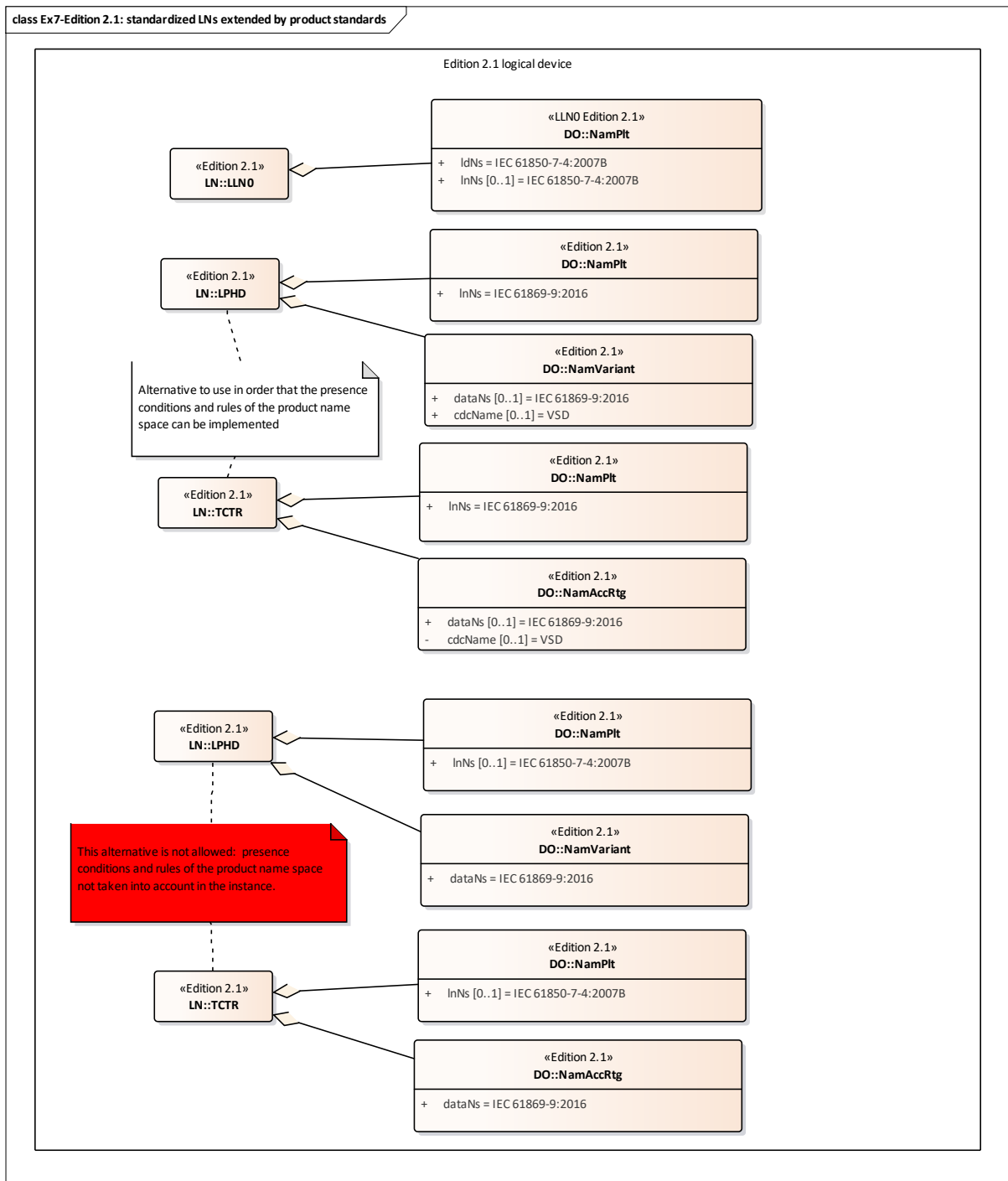


Figure J.8 – Edition 2.1 device: standardized LNs extended by other domains

J.9 Example 8 – Standardized logical nodes introduced by other domain applications

This clause gives examples about the use of logical nodes that have been introduced by the DER domain application. The DER domain application reuses logical node definitions found in the IEC 61850-7-4 namespace. As such, the IEC 61850-7-420 specification shall declare that it includes the IEC 61850-7-4 namespace. Figure J.9 and Figure J.10 show different versions of IEC 61850-7-420 devices and the related mandatory/optional namespace declarations.

NOTE LLN0.NamPit.InNs specifies the basic namespace the device is based on.

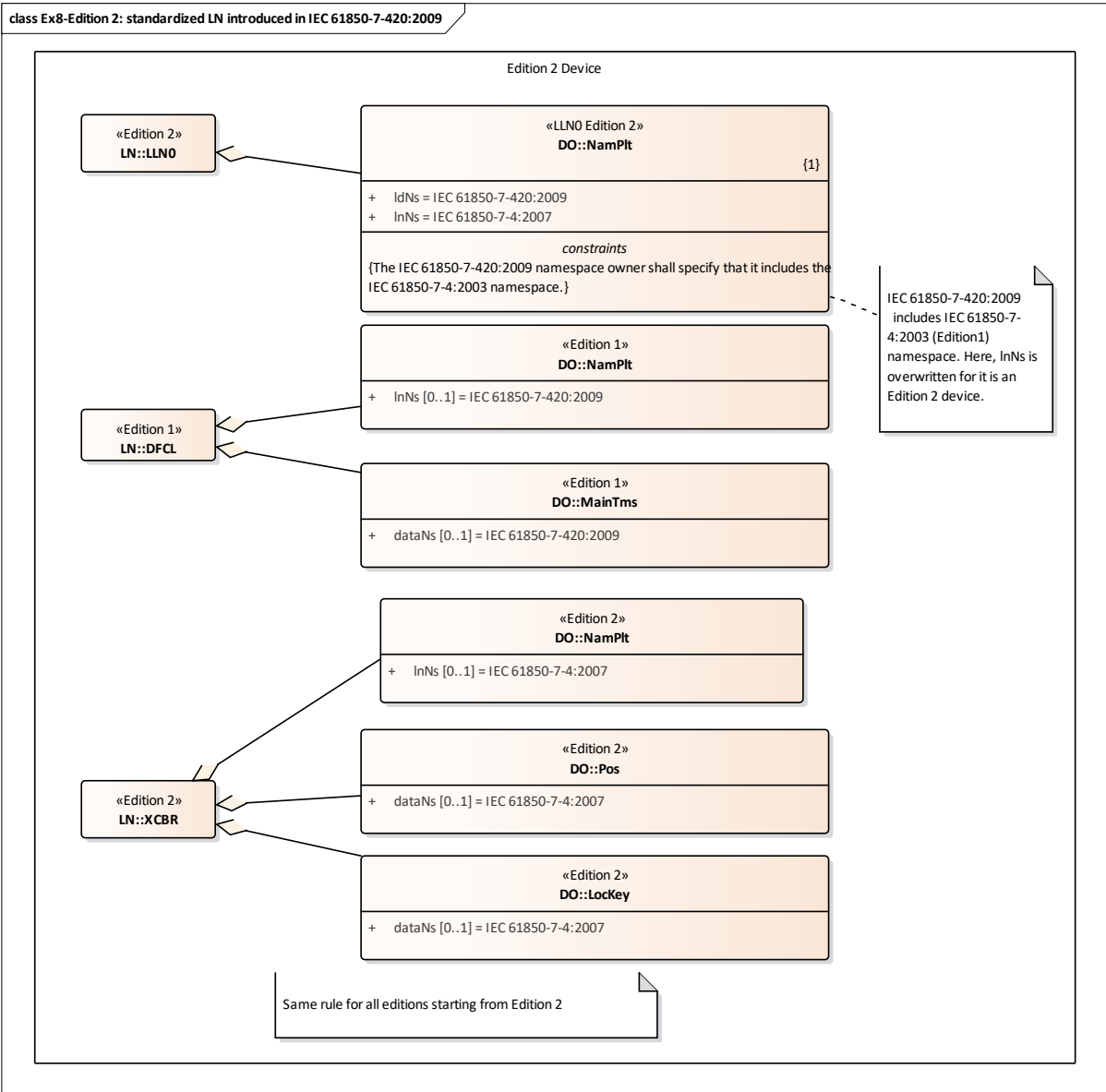


Figure J.9 – Edition 2 device: standardized LN introduced in IEC 61850-7-420:2009

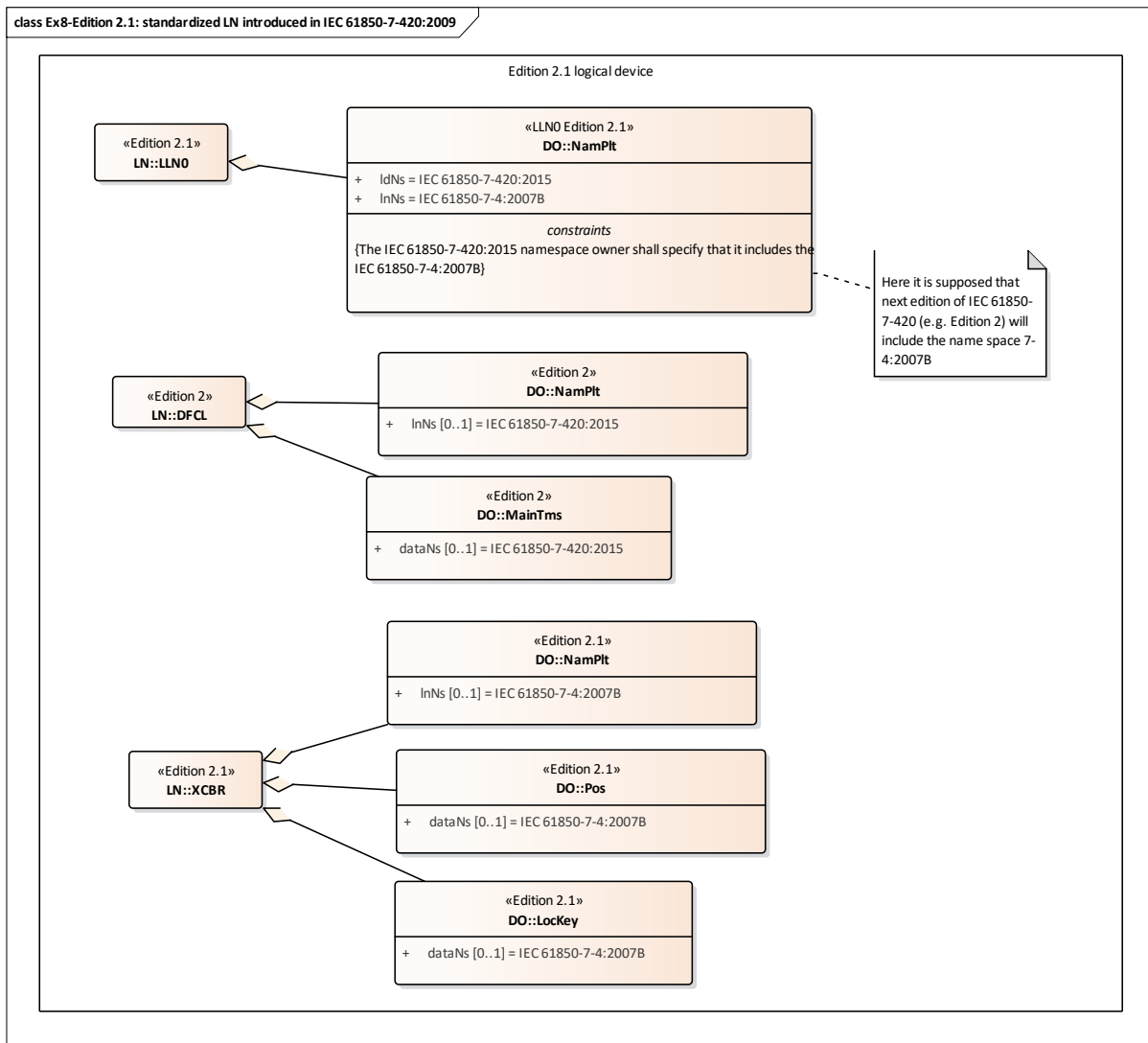


Figure J.10 – Edition 2.1 device: standardized LN introduced in IEC 61850-7-420:2015

NOTE The version for the second edition of the IEC 61850-7-420 is unknown at this time, and the 2015 is used as an example.

J.10 Example 9 – Standardized logical nodes introduced by other domain applications than moved to IEC 61850-7-4

This clause illustrates the situation where logical nodes first defined in a domain namespace (here IEC 61850-7-420) are then moved to IEC 61850-7-4. This situation is not privileged and should be covered by means of technical reports. However it cannot be completely avoided. The situation described is the following:

- The LN STMP has first been defined in IEC 61850-7-420:2009 than moved to IEC 61850-7-4:2007.
- IEC 61850-7-420:2009 defines a data object TmpRteSt (Temperature rate alarm) while IEC 61850-7-4:2007 does not.
- IEC 61850-7-4:2007B defines a data object for temperature rate alarm and have harmonized the name to RteAlm.

The examples given in Figure J.11 and Figure J.12 show different namespace versions of devices and the rules about namespace declarations.

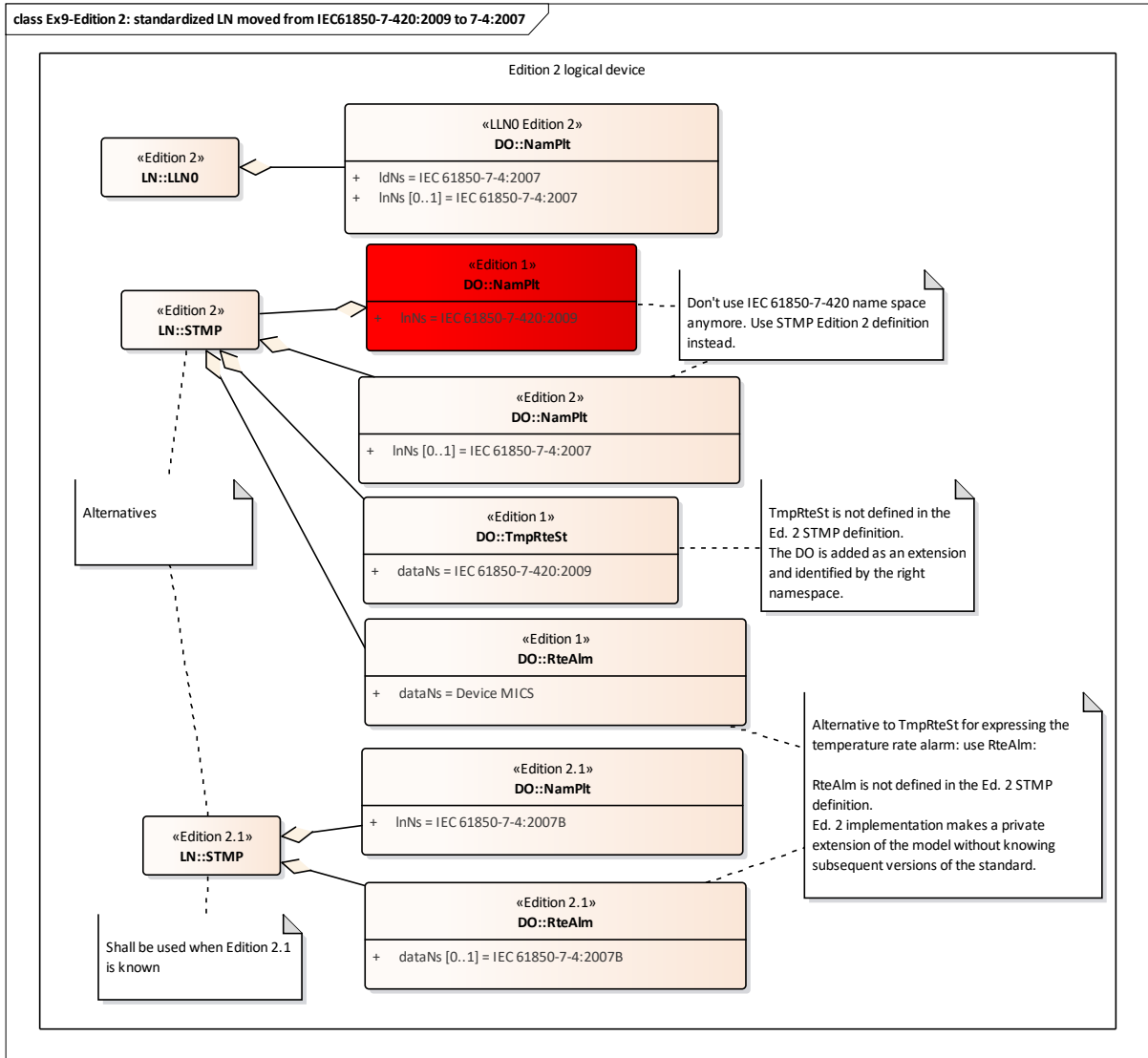


Figure J.11 – Edition 2 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007

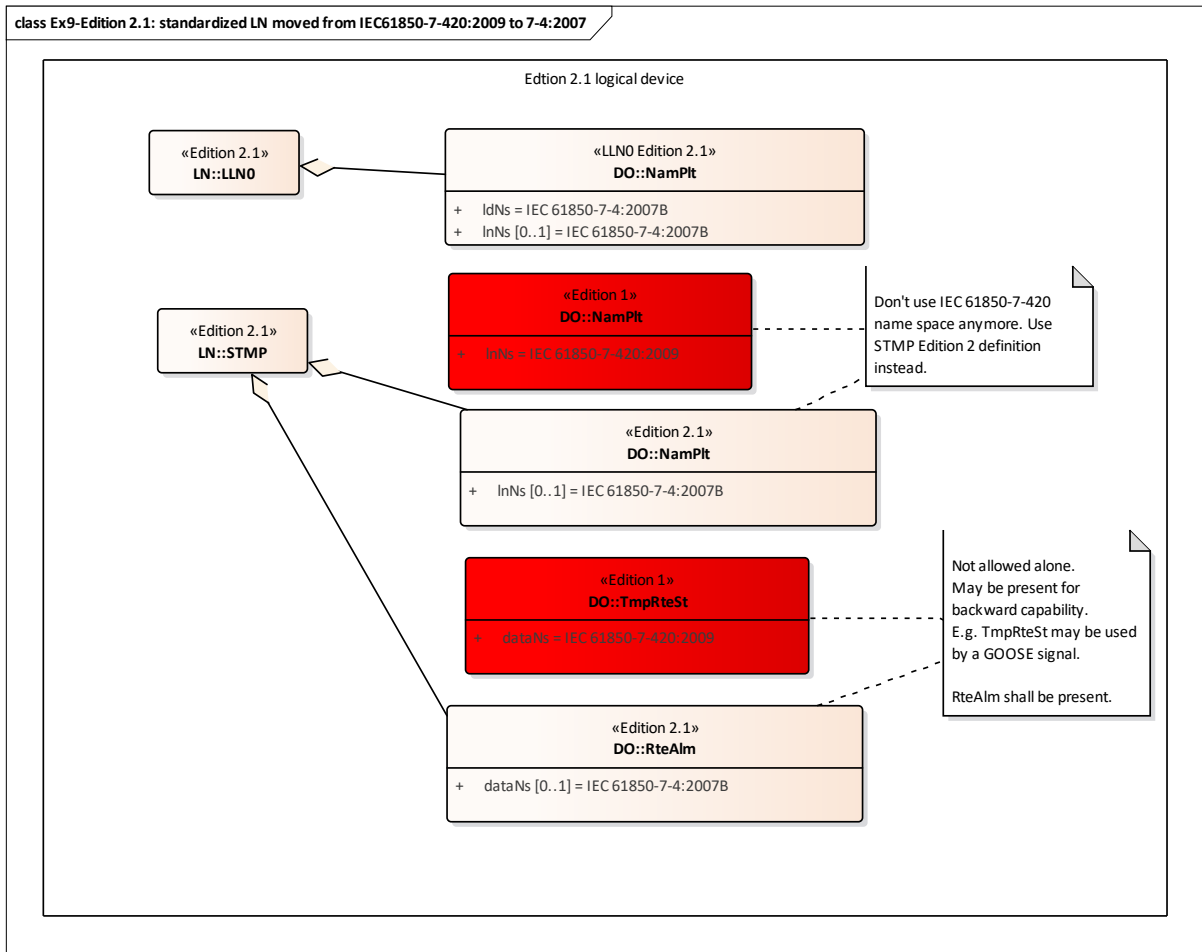


Figure J.12 – Edition 2.1 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007

Annex K (normative)

General requirements and recommendations regarding compatibility issues between different versions of IEC 61850

K.1 Overview

K.1.1 General

New versions and revisions of the standard are driven by

- Clarification on existing definitions: definitions or description may sometimes have been interpreted differently leading therefore to editorial tissues or interoperability tissues,
- Fixing errors: this is necessary if faults or inconsistencies exist. Corrections are required and must be performed, even if it endangers compatibility. Compatibility rules when required shall be standardized to guarantee a deterministic behaviour when deploying systems based on different revisions of the standard,
- Enhancements: adding new features to support new functionality. The introduction of new elements of the data model is driven by functional requirements discussed in Technical Reports.

This annex standardizes rules and associated behaviours following modification use cases to specify the expectation on implementations with regards to backward / forward compatibility of systems.

K.1.2 Definitions (to go to part 2):

Backwards compatibility (BC): an information user (claiming conformance to a version of the standard) can understand also information of an information provider (claiming conformance to an older version of the standard) without the need to apply a compatibility rule.

Generic rule: standardized requirement for tolerance regarding changes in the standard to be followed by configuration tools and IEDs.

NOTE 1 Generic rules are defined in Clause K.1.5.

Compatibility rule (CRu): standardized rule for implementation how to handle incompatible changes of the standard. They may refer to a generic rule.

NOTE 2 Compatibility rules are defined in the context of the use cases discussed in Clauses K.2 and K.3.

Specific compatibility rule: Compatibility rule that is defined in the context of an instance of a change.

NOTE 3 A specific compatibility rule can only be used for backwards compatibility, since it has to be defined in the context of an instance of a change, which is not known yet for forward compatibility.

NOTE 4 A specific compatibility rule will not be defined for the integration of model from a Technical Report resp. Technical Specification into an International Standard.

Forwards compatibility (FC): an information user (claiming conformance to a version of the standard) can understand also information from an information provider (claiming conformance to a newer version of the standard)

NOTE 5 The interpretation of “understand” is context specific and will be detailed in the context of the different parts of standard.

K.1.3 Introduction to compatibility discussion

This annex discusses compatibility related to the parts IEC 61850-7-2, IEC 61850-7-3 and IEC 61850-7-4 as well as generalized to all International Standard defining IEC 61850 Compatible Logical Node Classes. Subclause K.1.6 lists the use cases considered.

Compatibility is discussed from the perspective of the information user. Table K.1 provides an overview of information users and associated information providers.

Table K.1 – Information users and information providers

Information user	Information provider
Client	Server
Subscriber	Publisher
IED tool of information user	System tool, IED tool of information provider
System tool	IED tool of information provider, System tool (e.g. SED or SSD files)

NOTE For the version of tools, this annex refers to the version of the data model the tool is using. The SCL version supported by the tool may be independent from the data model version.

The impact on compatibility of different changes is defined by the following key words:

- **Guaranteed:** in any case possible
- **Achievable:** is possible through applying generic rules and / or compatibility rules.
- **Not achievable:** impossible
- **Not relevant:** the use case does not exist

NOTE The parts 7-x will maintain a history of the changes.

In some cases:

- Downgrading rules are provided for compatibility with IED tools and associated IEDs conformant to Edition 1 or Edition 2. If the compatibility rules specified in this annex are implemented with Edition 2.1 and forward, downgrading is usually not needed as long as there is no Edition 1 or Edition 2 involved.
- System design rules to be enforced by the system configuration tool are provided to avoid unnecessary updates.

K.1.4 Assumptions

K.1.4.1 Function related compatibility versus information exchange

It is important to differentiate between function related compatibility and information exchange related compatibility.

Since the standardization of functions is out of scope for IEC 61850, function related compatibility is not discussed in this annex. If new functionality introduced by later versions of the standard shall be used in a system (and hence, the semantic of the new elements added needs to be understood by the components), the components involved in that functionality may most likely require an update to support the latest version of the standard. Such functional impacts will not be considered in this annex.

However, in some cases, semantic changes of the data model may impact information exchange related functionality or other functionality defined in IEC 61850. These use cases have to be discussed on an individual case basis possibly requiring a specific compatibility rule for backwards compatibility. For forward compatibility, information users need to create a warning if they do not understand something and may prevent the usage depending on the case.

Example 1: An application creates an action based on a specific data object value. The application needs to understand the semantic of that data value. Such a case is outside the consideration.

Example 2: A gateway forwards a new DO; but it does not need to understand it. Such a case is considered.

Example 3: Changes in the attribute t are semantic changes; however, the functionality is defined in IEC 61850 and is related to information exchange. Such a case would be considered with a specific compatibility rule.

K.1.4.2 Modification related to service parameters and control block attributes

Further it has to be noted that modifications related to service parameters and control blocks only impact clients and subscribers. The impact of such modifications on tools is discussed in the compatibility Annex I of IEC 61850-6:2009/AMD1:2018 and is typically covered by the `scl mustUnderstand / mayIgnore` rules, as it usually results in SCL schema changes.

K.1.5 Generic rules

K.1.5.1 Tool design

Forward and backward compatibility of a tool of an information user requires, that the tool accepts a future or older (unknown) data model namespace in an IED section of an information provider and is able to analyse the structure of such a namespace. It shall as well be able to import SCL files with unknown basic types and ignore those, as long as it is not required to use them.

The list of the natively known core namespace(s) by the system tool, shall be expressed in a SICS.

K.1.5.2 Client

Forward and backward compatibility of a client, requires that the client is able to deal with unknown elements in a server data model. As a minimum, the client shall be able to ignore them. Additionally to ignore them, a client may still handle them and forward to the application (e.g. a gateway or an HMI). The capability of the client to do so shall be expressed in a PICS.

A client shall be designed such that even, if it expects an element to be mandatory according his version, it shall be tolerant to accept that the element is not there. It shall issue a warning to the user.

K.1.5.3 Deprecation and Removal

If something has been removed from the standard, it is not known any more and an implementation according a newer version will not know the element. If something has been deprecated, it is still known by the standard. Deprecated elements of the model will still be contained in the UML model of the standard. Elements that have been removed will only be mentioned in the compatibility Annex of the concerned part of the standard. An implementation according a newer version may claim in the PICS that it supports the deprecated element, but it is not required to do so.

The names of deprecated elements are reserved, i.e. cannot be re-used with another semantic in a future revision of the namespace. If elements from the data model related to functionality in the standard are removed or deprecated, specific compatibility rules may be required.

K.1.5.4 Semantic changes and extensions related to functionality defined in IEC 61850

In some cases, semantic changes of the data model may impact information exchange related functionality or other functionality defined in IEC 61850. In that case, forward compatibility is not achievable. The information users shall issue in such cases a warning to the user and may prevent the usage depending on the case.

An example for that would be a new control model which results in a new enumerated value for `ctlModel` that the client cannot serve.

K.1.6 Overview on use cases

K.1.6.1 General

This subclause contents the use cases of the changes. In reality, a change in the standard can be a superposition of several use cases. Example: adding a new DO in an existing LN using a new CDC with mandatory DA using a new basic type superposes the use cases:

- using a new types
- add a new CDC
- extend existing LN with new DO.

Here, the use case using new type determines the expected compatibility set. The part specific lists of the modification to consider are not explicit because a strong context in another part may have already settled the case. In the above mentioned example using a new type in part 7-2 will lead to a table of impacted CDCs in part 7-3 but not to a table of impacted DO in part 7-4.

K.1.6.2 Use cases on the data model

Table K.2 gives data model use cases.

Table K.2 – Data model use cases

Use Case Number		Use Case Description	Part
		<i>Addition / Extension of the model</i>	
1		<i>Using a new basic or non-structured type</i>	
1	a	Add new type	7-2, 8-x, 9-x
1	b	Extend existing CDC with DA of a new type and existing FC	7-3
1	c	Extend existing CDC with DA of a new type and new FC	7-3
1	d	Add a new CDC with DA of a new type and existing FC	7-3
1	e	Add a new CDC with DA of a new type and new FC	7-3
1	f	Extend existing constructed DA type with Sub DA of new type	7-3
2		<i>Using a new FC using existing types</i>	
2	a	Add a new FC	7-2, 8-x
2	b	Extend existing CDC with DA of new FC	7-3
2	c	Add a new CDC with DA of a new FC	7-3
3		<i>Add elements of existing types, existing FC</i>	
3	a	Add DataAttribute (DA) to a CDC	7-3
3	b	Add SubDataAttribute (Sub DA) to a constructed DA type	7-3
3	c	Add SubDataObject (Sub DO) to a CDC	7-3
3	d	Add a new constructed DA type	7-3

Use Case Number		Use Case Description	Part
3	e	Add DataAttribute (DA) of a new Enumeration type	7-3
4		<i>Using new CDC based on existing types, existing FCs</i>	
4	a	Add a new CDC	7-3
4	b	New DO / Data Object Class based on new CDC in existing LN	7-4
4	c	New LN with DO / DataObject Class based on new CDC	7-4
5		<i>Using new DOs of existing CDCs</i>	
5	a	Extend existing LN with new DO	7-4
5	b	Extend existing LN with existing Data Object Class	7-4
5	c	New Data Object Class / New DO of an Enumerated CDC using new enumeration List	7-4
5	d	Add an existing Data Object Class in a New LN	7-4
5	e	New Data Object Class / New DO in a new LN	7-4
		<i>Change name of elements</i>	
6		<i>Rename a DO</i>	7-4
7		<i>Rename a DA, Sub DO or Sub DA</i>	
7	a	Rename a DA	7-3
7	b	Rename a Sub DO	7-3
7	c	Rename a Sub DA	7-3
		<i>Removing / deprecating / change type elements</i>	
8		Deprecation of a FC	7-2
9		Deprecation of a DA	7-3
10		Removal of a DA	7-3
		<i>Change in Logical Node</i>	
11	a	Deprecation of a DO	7-4
11	b	Removal of a DO	7-4
11	c	Deprecation of an LN	7-4
11	d	Removal of an LN	7-4
		<i>Changing type of element</i>	
		<i>Changing presence conditions</i>	
12		<i>Use a weaker Presence Condition</i>	
12	a	Presence Condition of a DA	7-3
12	b	Presence Condition of a Sub DA	7-3
12	c	Presence Condition of a Sub DO	7-3
12	d	Presence Condition of a DO	7-4
13		<i>Use a stronger Presence Condition</i>	
13	a	Presence Condition of a DA	7-3
13	b	Presence Condition of a Sub DA	7-3
13	c	Presence Condition of a Sub DO	7-3
13	d	Presence Condition of a DO	7-4
		<i>Modifications with enumerated types</i>	
14		<i>Extend existing Enumeration List with an Enumerated value</i>	7-2, 7-3, 7-4
15		<i>Modification of an Enumerated Value</i>	7-2, 7-3, 7-4
16		<i>Deprecation of an Enumerated Value</i>	7-2, 7-3, 7-4

Use Case Number	Use Case Description	Part
	<i>Modifications with domain types</i>	
17	<i>Extension of a PACKED LIST</i>	7-2, 8-x, 9-x
18	<i>Extension of ObjectName / ObjectReference</i>	7-2, 8-x, 9-x
18	a Extend LDName	7-2
18	b Extend LNName	7-2
18	c Extend DataObjectName	7-2
18	d Extend an ObjectReference type	7-2
18	e Extend ControlBlockName	7-2
18	f Extend a length of a control block attribute of type String	7-2

K.1.6.3 Use cases related to the services

Table K.3 gives services use cases.

Table K.3 – Services use cases

Use Case Number	Use Case Description	Part
30	<i>Using new type</i>	
30	a Extend control block class with an attribute of a new type	7-2, 8-x, 9-x
30	b Extend a service with a parameter of a new type	7-2, 8-x, 9-x
30	c Add a service with a parameter of a new type	7-2, 8-x, 9-x
30	d Add a new control block class with an attribute of new type	7-2, 8-x, 9-x
31	<i>Add new control block class in GenLogicalNodeClass</i>	7-2, 8-1, 9-x
	<i>Using new Services</i>	
32	<i>Add a service with parameters of existing types</i>	7-2, 8-x, 9-x
	<i>Extending Control Block Class with existing types</i>	
33	a Extend control block class with an attribute of existing type	7-2, 8-x, 9-x
33	b Extend a service with a parameter of existing type	7-2, 8-x, 9-x
33	c Extend Tracking CDC with an attribute of a existing type	7-3
	<i>Change name of elements</i>	
34	<i>Rename a service parameter not associated with a control block attribute</i>	7-2
	<i>Removing/ Deprecating Control Block Classes</i>	
35	a Deprecate a control block class	7-2, 8-x, 9-x
35	b Removing a control block class	7-2, 8-x, 9-x
	<i>Changing presence condition</i>	
	<i>Use a stronger Presence Condition</i>	
36	a Presence Condition of a Control Block attribute	7-2, 8-x, 9-x

Use Case Number		Use Case Description	Part
36	b	Presence Condition of a service parameter	7-2, 8-x, 9-x

K.1.6.4 Forbidden cases in the future

The use cases listed in Table K.4 are forbidden in future but some of them have been used in the past to correct initial design errors in the standard. The dedicated parts will list the instances where use cases from Table K.4 have been applied and will discuss the consequences for compatibility on an individual basis.

Table K.4 – Data Forbidden cases

Use Case Number		Use Case Description	Part
f1		Deprecation of a type	7-2
f2		Removal of a type	7-2
f3		Removal of a functional constraint	7-2
f4		Rename a Logical Node Class	7-2
f5		Deprecate/Remove control block attribute	7-2
f6		Deprecate/Remove control block service	7-2
		<i>Renaming and control blocks</i>	
f7	a	Rename a control block attribute	7-2
f7	b	Rename a service parameter associated with a control bloc attribute	7-2
		<i>Weaker presence condition and control blocks</i>	
f8	a	Use a weaker presence condition of a control block attribute	7-2
f8	b	Use a weaker presence condition of a service	7-2
		<i>Enumerated Value</i>	
f9	a	Removal of an enumerated value	7-2, 7-3, 7-4
f9	b	Change a semantic of an enumerated value	7-2, 7-3, 7-4
f10		Changing CDC of a DO	7-4
f11		Extension of a CODED enumerated type	7-2
		<i>Changing type of element</i>	
f12	a	Change a type of a Control Block attribute	7-2, 8-x, 9-x
f12	b	Change a type of a Service Parameter	7-2, 8-x, 9-x
f13		Change type of a DA	7-3
f14		Deprecation of a CDC	7-3
f14		Shorten name length	7-2

Forbidden cases may be used while integrating model from Technical Reports or Technical Specifications into International Standards.

K.2 Use cases related to the data model

K.2.1 Use case 1: Add new type

A new type may be added to introduce new functionality. Table K.5 shows the impacts/recommendations related to the use case.

Table K.5 – Add new basic type

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> Not achievable if the element with the new type has to be processed by the client (e.g. within a member of a report). Otherwise achievable</p> <p>Compatibility rule: The client, while browsing a data model, shall be able to ignore elements of an unknown type.</p> <p><i>Backwards compatibility:</i> Guaranteed, if DA is optional; achievable if extension DA is mandatory</p> <p>Requires a specific compatibility rule if DA is mandatory</p>
Subscriber	<p><i>Forward compatibility:</i> Not achievable.</p> <p><i>Backwards compatibility:</i> Guaranteed.</p>
IED tool of information user	<p><i>Forward compatibility:</i> Not achievable, if the IED associated to the tool has to process the element with the new type; otherwise guaranteed.</p> <p><i>Backwards compatibility</i> Guaranteed</p>
System tool	<p><i>Forward compatibility:</i> Not achievable if the system configuration requires data flow engineering involving element with the new type; otherwise guaranteed</p> <p>A tool shall not support data flow engineering involving elements including unknown basic types,</p> <p><i>Backwards compatibility:</i> Guaranteed</p>
System design rule	<p>To avoid unnecessary update, data of the new types shall be used only in datasets that are sent to IEDs that can use and process them properly</p>

This change is not forward compatible as receiver compliant with former version of the standard cannot process the new basic type. An update of the receiver is required as soon as the system configuration requests that the receiver (client/subscribers) is able to receive data attribute of the new basic type. (Example: The INT64 basic type has been introduced in the second version of the standard. Receiver implementation compliant only with the first version of the standard is not expected to process the INT64 basic type and associated data attributes).

If a new DA (of the new type) is mandatory, backwards compatibility from the client perspective is not necessarily guaranteed, as the client may expect the DA to be present. In such a case, a special compatibility rule may be provided.

Forward compatibility of the tool is guaranteed (as long as the tool does not need to process the element) through the generic rule that a tool needs to be able to import SCL files with unknown data types (see Clause 175).

K.2.2 Use case 2: Add new FC

Extension of a common data class with a DA using a new functional constraint is used for introducing new functionality. Table K.6 shows the impacts/recommendations related to the use case.

Table K.6 – Extend CDC with DA of new FC

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC.</p> <p>Compatibility rule: The client shall be able to ignore DAs with unknown FC</p> <p><i>Backwards compatibility:</i> guaranteed, if DA is optional; achievable if extension DA is mandatory</p> <p>Requires a specific compatibility rule if DA is mandatory</p>
Subscriber	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: the IED tool or the SW of the subscribing IED shall analyse the data structure and hence configure the subscriber to handle that new DA in the message</p> <p><i>Backwards compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> guaranteed</p> <p>NOTE: A system tool is able to import any SCL files that contain DOs with CDCs extended with new DAs of existing basic type.</p> <p><i>Backwards compatibility:</i> guaranteed</p>
Downgrading	<p>For IED tools and associated IEDs conformant to Edition 1 or Edition 2, to support forward compatibility for subscribers, a system tool shall ensure that new DAs are not included in datasets for messages that shall be sent to those IEDs, if the IED of the former version does not need to receive that information.</p>

If a new DA is mandatory, backwards compatibility from the client perspective is not necessarily guaranteed, as the client may expect the DA to be present. In such a case, a special compatibility rule may be provided.

K.2.3 Use case 3: Extend CDC with elements of existing types and FCs

Extension of an existing common data class with new elements using existing types is used for fixing errors and simplifying the variance of implementation compliant to the new version of the standard. Table K.7 shows the impacts/recommendations related to the use case.

Table K.7 – Add new DA using existing type and FCs

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC.</p> <p><i>Backwards compatibility:</i> guaranteed, if DA is optional; achievable if DA is mandatory</p> <p>Requires a specific compatibility rule if DA is mandatory</p>
Subscriber	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: the IED tool or the SW of the subscribing IED shall analyse the data structure and hence configure the subscriber to at least be able to ignore that new DA in the message</p> <p><i>Backwards compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: the IED tool shall use ExtRef.intAddr empty to expose that a binding of an unknown DA is not supported.</p> <p><i>Backwards compatibility:</i> guaranteed</p>

Concerned entity	Impact / recommendations
System tool	<p><i>Forward compatibility:</i> guaranteed</p> <p>NOTE: A system tool is able to import any SCL files that contain DOs with CDCs extended with new DAs of existing basic type and FC.</p> <p><i>Backwards compatibility:</i> achievable.</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user.</p>
Downgrading	<p>For IED tools and associated IEDs conformant to Edition 1 or Edition 2, to support forward compatibility for subscribers, a system tool shall ensure that new DAs are not included in datasets for messages that shall be sent to those IEDs, if the IED of the former version does not need to receive that information.</p>

If a new DA is mandatory, backwards compatibility from the client perspective is not necessarily guaranteed, as the client may expect the DA to be present. In such a case, a special compatibility rule may be provided.

K.2.4 Use case 4: Add a new DataObject (DO) based on new CDCs

New CDCs are introduced to add new functionality. The use case is discussed from the perspective of IEC 61850-7-3 assuming that the new CDC is using only existing FC and types. Table K.8 shows the impacts/recommendations related to the use case.

Table K.8 – Add a new DO based on new CDC using existing types and FCs

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC and the FC to understand what services can be applied to a DA.</p> <p><i>Backwards compatibility:</i> not relevant</p>
Subscriber	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: the IED tool or the SW of the subscribing IED shall analyse the data structure and hence configure the subscriber to at least be able to ignore elements of DOs based on that new CDC in the message.</p> <p><i>Backwards compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> achievable.</p> <p>Compatibility rule: the IED tool shall use ExtRef.intAddr empty to expose that a binding of a DO based on that new CDC is not supported.</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
Downgrading	<p>For IED tools and associated IEDs conformant to Edition 1 or Edition 2, to support forward compatibility for subscribers, a system tool shall ensure that DOs based on new CDCs are not included in datasets for messages that shall be sent to those IEDs, if the IED of the former version does not need to receive that information.</p>

For the use cases 4b and 4c, if the CDC uses new FCs or new types, the use cases 1 or 2 need to be considered with regard to compatibility.

K.2.5 Use case 5: Add a new DO based on existing CDCs

Addition of a new LN class is used for introducing new functionality. Addition of DO as optional or mandatory is used for modification or completion of existing functionality. Table K.9 shows the impacts/recommendations related to the use case.

Table K.9 – Adding a new DO based on existing CDC

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Guaranteed <i>Backwards compatibility:</i> Not relevant
Subscriber	<i>Forward compatibility:</i> Guaranteed. <i>Backward compatibility:</i> Not relevant
IED tool of information user	<i>Forward compatibility:</i> Guaranteed, <i>Backwards compatibility:</i> Not relevant
System tool	<i>Forward compatibility:</i> Guaranteed <i>Backward compatibility:</i> Not relevant

Forward compatibility is guaranteed in the context of IEC 61850, as adding DOs is relevant for the application function, which is out of the scope for this discussion (see K.1.3).

K.2.6 Use case 6: Rename a DO

Renaming of DO is used to correct the standard, e.g. to keep uniqueness of {DOName, CDC Type}. Table K.10 shows the impacts/recommendations related to the use case.

Table K.10 – Renaming a DO

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.
Subscriber	<i>Forward compatibility:</i> Guaranteed. <i>Backward compatibility:</i> Guaranteed
IED tool of information user	<i>Forward compatibility:</i> Guaranteed, <i>Backwards compatibility:</i> Guaranteed
System tool	<i>Forward compatibility:</i> Guaranteed <i>Backward compatibility:</i> Guaranteed

Compatibility is guaranteed as the DO instance using the older name (backwards compatibility) or the newer name (forward compatibility) is considered as an unknown DO. The DO with the newer name (backwards compatibility) or the older name (forward compatibility) is considered as a removed DO.

K.2.7 Use case 7: Rename a DA, subDO or subDA

Renaming DA, subDO or subDA is used to make corrections to the standard. Table K.11 shows the impacts/recommendations related to the use case.

Table K.11 – Renaming a DA, subDO or subDA

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC and the FC to understand what services can be applied to a DA.</p> <p><i>Backwards compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the old structure of the CDC and the FC to understand what services can be applied to a DA.</p> <p>If DA is mandatory, a specific compatibility rule may be required</p>
Subscriber	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: the IED tool or the SW of the subscribing IED shall analyse the data structure and hence configure the subscriber to at least be able to ignore that renamed DA in the message. A binding by the system tool based on type can improve the compatibility.</p> <p><i>Backward compatibility:</i> achievable</p> <p>Requires a specific compatibility rule.</p>
IED tool of information user	<p><i>Forward compatibility:</i> achievable.</p> <p>Compatibility rule: the IED tool shall use ExtRef.intAddr empty to expose that a binding of an renamed (unknown) DA is not supported.</p> <p><i>Backwards compatibility:</i> achievable</p> <p>Requires a specific compatibility rule</p>
System tool	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user.</p> <p><i>Backward compatibility:</i> achievable</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user</p>
Downgrading rule	<p>For IED tools and associated IEDs conformant to Edition 1 or Edition 2, to support forward compatibility, a system tool may export a SCD file in an old version for subscribers that understand the old version and perform a downgrading by replacing the DA name of the new version of the namespace by the name that was used in the older version of the namespace.</p> <p>NOTE This is only useful for tools that use the SCL model; it will not help a client that browses the model</p>

If the renamed DA was mandatory, forward compatibility for the client is still possible, as it has to deal with the situation, that an expected mandatory DA is not present anymore. See K.1.5.2). Backwards compatibility can be achieved with a specific compatibility rule.

K.2.8 Use case 8: Deprecation of a FC

Deprecation of a FC may be applied if the functionality is not needed anymore or if a better approach has been defined to model the functionality. Table K.12 shows the impacts/recommendations related to the use case.

Table K.12 – Deprecation of a FC

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> guaranteed as long as none of the DAs removed with the FC have been mandatory. If at least one of the deprecated DA was mandatory, achievable with generic rule that client shall be tolerant for missing elements (see K.1.5)</p> <p><i>Backwards compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition to learn the old structure of the CDC and how to ignore the DAs contained in the deprecated FC</p>
Subscriber	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backward compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> guaranteed if DAs with deprecated FC were only optional, achievable if at least one DA was mandatory</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user</p> <p><i>Backward compatibility:</i> guaranteed</p>

Backwards compatibility can be achieved if the client declares support of deprecated elements.

K.2.9 Use case 9: Deprecation of a DA

Deprecation of a DA may be applied if the functionality is not needed anymore or if a better approach has been defined to model the functionality. Table K.13 shows the impacts/recommendations related to the use case.

Table K.13 – Deprecation of a DA

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> guaranteed if DA was optional, achievable with generic rule that client shall be tolerant for missing elements (see K.1.5)</p> <p><i>Backwards compatibility:</i> guaranteed, if the client still supports the deprecated DA, otherwise achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the old structure of the CDC and the FC to understand what services can be applied to a DA.</p>
Subscriber	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backward compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> guaranteed if DA was optional, achievable if DA was mandatory</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user</p> <p><i>Backward compatibility:</i> guaranteed</p>

K.2.10 Use case 10: Removal of a DA

Removal of a DA may be applied if the functionality is not needed anymore and there is no reason that newer devices can still understand the old version. Table K.14 shows the impacts/recommendations related to the use case.

Table K.14 – Removal of a DA

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> guaranteed if DA was optional, achievable with generic rule that client shall be tolerant for missing elements (see clause K.1.5)</p> <p><i>Backwards compatibility:</i> achievable</p> <p>Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the old structure of the CDC to understand how to ignore the unexpected DA</p>
Subscriber	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backward compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> guaranteed if DA was optional, achievable if DA was mandatory</p> <p>Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user</p> <p><i>Backward compatibility:</i> guaranteed</p>

K.2.11 Use case 11: Deprecation of a DO

Deprecation of a DO may happen, if the function is not needed anymore or if the modelling has been improved. Table K.15 shows the impacts/recommendations related to the use case.

Table K.15 – Deprecation of a DO

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable with generic rule that client shall be tolerant for missing elements (see clause K.1.5)</p> <p><i>Backwards compatibility:</i> guaranteed</p>
Subscriber	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backward compatibility:</i> guaranteed</p>
IED tool of information user	<p><i>Forward compatibility:</i> guaranteed</p> <p><i>Backwards compatibility:</i> guaranteed</p>
System tool	<p><i>Forward compatibility:</i> achievable with generic rule that client shall be tolerant for missing elements (see clause K.1.5)</p> <p><i>Backward compatibility:</i> guaranteed</p>

K.2.12 Use case 12: Use a weaker presence condition

Changing the presence condition may be done for corrections. Table K.16 shows the impacts/recommendations related to the use case.

Table K.16 – Use of a weaker presence condition

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> achievable with generic rule that client shall be tolerant for missing elements (see clause K.1.5) <i>Backwards compatibility:</i> guaranteed
Subscriber	<i>Forward compatibility:</i> guaranteed <i>Backward compatibility:</i> guaranteed
IED tool of information user	<i>Forward compatibility:</i> guaranteed <i>Backwards compatibility:</i> guaranteed
System tool	<i>Forward compatibility:</i> achievable. Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user. <i>Backward compatibility:</i> guaranteed

K.2.13 Use case 13: Use a stronger presence condition

Changing the presence condition may be done for corrections. Table K.17 shows the impacts/recommendations related to the use case.

Table K.17 – Use of a stronger presence condition

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> guaranteed <i>Backwards compatibility:</i> achievable with generic rule that client shall be tolerant for missing elements (see clause K.1.5)
Subscriber	<i>Forward compatibility:</i> guaranteed <i>Backward compatibility:</i> guaranteed
IED tool of information user	<i>Forward compatibility:</i> guaranteed <i>Backwards compatibility:</i> guaranteed
System tool	<i>Forward compatibility:</i> guaranteed <i>Backward compatibility:</i> achievable. Compatibility rule: The system tool shall be tolerant, if e.g. a mandatory CF attribute it tries to configure is not there anymore, but it shall issue a warning to the user.

K.2.14 Use case 14: Extend Enumeration List with an enumerated value

Enumeration lists are extended for adding new functionality. Table K.18 shows the impacts/recommendations related to the use case.

Table K.18 – Extend an enumeration with an enumerated value

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> achievable if the enumeration does not affect information exchange; otherwise not achievable. Compatibility rule: The client shall be tolerant for unknown enumerated values <i>Backwards compatibility:</i> Guaranteed.
Subscriber	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.
IED tool of information user	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.
System tool	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.

For forward compatibility see K.1.5.4. With regard to backwards compatibility, a new client that tries to write a value not know by the old server will receive a negative response.

Enumeration are per IEC 61850-7-2 extendable. However interoperability at application level may not be achievable. If interoperability with automation function is expected, then a firmware update may be required. Examples are:

- A reported status value (7-4, 7-3, 7-2) can be used to HMI/Logic/archive purpose.
- A control or setting value (7-4) can be used by a user; the user being responsible for the interpretation of the semantic.
- A new configuration value (7-2, 7-3) can be used by a user; the user being responsible for the interpretation of the semantic.

K.2.15 Use case 15: Modification of an Enumerated value

Literal associated to an ord Enumerated value can be modified because of correction or clarification of the value. The semantic remains unchanged. New semantics are introduced with extensions of Enumeration list. Table K.19 shows the impacts/recommendations related to the use case.

Table K.19 – Modification of an enumerated value from an enumeration list

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Guaranteed. <i>Backward compatibility:</i> Guaranteed
Subscriber	<i>Forward compatibility:</i> Guaranteed: ord is the transmitted value. <i>Backwards compatibility:</i> Guaranteed
IED tool of information user	<i>Forward compatibility:</i> Achievable Compatibility rule: Changes of literal shall be detected by the IED tool as both literals share the same ord in the EnumType definition as part of the DataTypeTemplate. <i>Backwards compatibility:</i> achievable Compatibility rule: Changes of literal shall be detected by the IED tool as both literals share the same ord in the EnumType definition as part of the DataTypeTemplate.
System tool	<i>Forward compatibility:</i> Achievable. Compatibility rule: The system tool shall use the literal as defined by the IED <i>Backwards compatibility:</i> Achievable. Compatibility rule: The system tool shall use the literal as defined by the IED

K.2.16 Use case 16: Deprecation of an Enumerated value

Deprecation of enumerated values is done for corrections or if the functionality is not needed anymore. Table K.20 shows the impacts/recommendations related to the use case.

Table K.20 – Deprecate an enumerated value from an enumeration list

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed if the DA does not affect information exchange; otherwise achievable using a specific compatibility rule (e.g. support of a deprecated control model).
Subscriber	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.
IED tool of information user	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed,
System tool	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.

Forward compatibility is guaranteed since in any case an information provider may limit the valid enumerations to a subset. So deprecating an enumerated value provides the same issues as if an information provider does not support a value.

K.2.17 Use case 17: Extension of a PACKED LIST

Extension of a PACKED LIST has impact on the SCSM. Extension(s) may be possible within a given range without impacting the SCSM. An extension of a PACKED LIST shall only appear if the forward/backward compatibilities of the SCSM is guaranteed; otherwise, a new basic type shall be added to the model. Table K.21 shows the impacts/recommendations related to the use case.

Table K.21 – Extension of a PACKED LIST

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Guaranteed. The unknown bits are ignored. <i>Backwards compatibility:</i> Achievable. Specific compatibility rule is required to default the not received bits.
Subscriber	<i>Forward compatibility:</i> Guaranteed. The unknown bits are ignored. <i>Backwards compatibility:</i> Achievable. Specific compatibility rule is required to default the not received bits.
IED tool of information user	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.
System tool	<i>Forward compatibility:</i> Guaranteed. <i>Backwards compatibility:</i> Guaranteed.

K.2.18 Use case 18: Extend Name Length

The second version of the standard has extended references from 65 to 129 characters. This change is partly backward compatible as names are a configuration issue and objects that need to be transmitted to receivers compliant with previous revisions of the standard may be configured with shorter names. Additionally, the service capability exposes the supported

reference and naming size to allow configuration to implement a compatibility rule. Table K.22 shows the impacts/recommendations related to the use case.

Table K.22 – Extend size of names / references

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> not achievable – an update of the client is required as soon as objects with longer name are to be received or the client is using the discovery services, otherwise achievable using the compatibility rule at the system tool. <i>Backwards compatibility:</i> guaranteed.
Subscriber	<i>Forward compatibility:</i> not achievable – an update of the subscriber may be required as soon as objects with longer name are to be received, otherwise achievable using the compatibility rules at the system tool. <i>Backwards compatibility:</i> guaranteed.
IED tool of information user	<i>Forward compatibility:</i> not achievable: if the IED associated to the tool needs to be upgraded, the tool needs to be upgraded as well. <i>Backwards compatibility:</i> guaranteed.
System tool	<i>Forward compatibility:</i> not achievable – SCL files including too long names will not be compliant to the schema. Objects will be best case ignored; worst case IID/ICD refused (obviously new SCL schema version) – upgrade of the tool is required SED import: object with too long name are ignored or not even imported (downgrading rules). Data flow configuration only possible with object with short names. <i>Backwards compatibility:</i> achievable – as both long and shorter names are supported. Compatibility rule: The system shall respect the capability of client/subscriber indicated by the Services.nameLength attribute and forbid the data flow engineering involving name which are longer than the supported length scheme.

K.3 Use cases related to the services

K.3.1 Use case 30: Add new type

A new type may be added to introduce new functionality. New service parameter / control block attribute of a new type impact also the associated tracking CDC. Use case 1 therefore applies in addition for tools. Table K.23 shows the impacts/recommendations related to the use case.

Table K.23 – Add new basic type

Concerned entity	Impact / recommendations
Client	<i>Forward compatibility:</i> Not achievable. <i>Backwards compatibility:</i> Guaranteed, if service parameter is optional; achievable if service parameter is mandatory. Requires a specific compatibility rule if service parameter is mandatory.
Subscriber	<i>Forward compatibility:</i> Not achievable. <i>Backwards compatibility:</i> Guaranteed, if service parameter is optional; achievable if service parameter is mandatory. Requires a specific compatibility rule if service parameter is mandatory.

This change is not forward compatible as receiver compliant with former version of the standard cannot process the new basic type. An update of the receiver is required as soon as the system configuration requests that the receiver (client/subscribers) is able to receive the service parameter of the new basic type.

If the new service parameter (of the new type) is mandatory, a special compatibility rule shall be provided.

K.3.2 Use case 31: Using new control block class

Addition of new control block classes is done for introducing and realizing new communication requirements. New control block classes will also imply new CDC (UC 4) for realizing the associated service tracking functionality. Table K.24 shows the impacts/recommendations related to the use case.

Table K.24 – Adding new Control Blocks

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client shall ignore the new control blocks classes that it may have discovered using the GetLogicalNodeDirectory.</p> <p><i>Backwards compatibility:</i> guaranteed.</p>
Subscriber	<p><i>Forward compatibility:</i> not relevant</p> <p><i>Backwards compatibility:</i> not relevant</p>

K.3.3 Use case 32: Using new services

Addition of new services is done for introducing and realizing new communication requirements. Support of optional services are declared in the PICS (e.g. the optional **GetMsvReference** was introduced in the first amendment of the second edition of the standard). New services shall not be mandatory; however they can be conditional to e.g. new control block classes. Table K.25 shows the impacts/recommendations related to the use case.

Table K.25 – Adding new services

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> not relevant</p> <p><i>Backwards compatibility:</i> guaranteed</p>
Subscriber	<p><i>Forward compatibility:</i> not achievable</p> <p><i>Backwards compatibility:</i> guaranteed. New services are never mandatory, and their support shall declared by SCL IED Capabilities, PICS or with Data Object visibility (e.g. LPHD.Sim). An SCSM service error shall indicate to the client that the service is not supported.</p>
IED tool of service caller	<p><i>Forward compatibility:</i> not achievable</p> <p><i>Backwards compatibility:</i> guaranteed.</p>
System tool	<p><i>Forward compatibility:</i> Not relevant.</p> <p><i>Backwards compatibility:</i> Not relevant.</p>
Downgrading	Not relevant.

Backward compatibility is guaranteed as new services are never mandatory, and their support shall be declared by SCL IED Capabilities, PICS or with Data Object visibility. An SCSM service error shall indicate to the client that the service is not supported.

K.3.4 Use case 33: Extending control block class with an attribute of existing type

Extensions of control blocks with new attributes are driven for enhancing the existing interaction with the control block (introduction of **BRCB.ResvTms** in the second edition of this standard) or for enhancing the description behind the control block (introduction of **SGCB.ResvTms** in the second edition of this standard). Table K.26 shows the impacts/recommendations related to the use case.

Table K.26 – Add new attributes in control block

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable</p> <p>Compatibility rule: The client ignores the new control block attributes it discovers using the Get<ControlBlock>Value().</p> <p><i>Backwards compatibility:</i> guaranteed, if control block attribute is optional; achievable if control block attribute is mandatory</p> <p>Requires a specific compatibility rule if control block attribute is mandatory.</p>
Subscriber	<p><i>Forward compatibility:</i> not achievable as soon as the control block extension has an impact on the publish functionality, and its usage is mandatory or required by the system specification, otherwise guaranteed.</p> <p><i>Backwards compatibility:</i> guaranteed, if control block attribute is optional; achievable if control block attribute is mandatory</p> <p>Requires a specific compatibility rule if control block attribute is mandatory.</p>

NOTE The client can use the Get<ControlBlock>Value() to identify if the new attribute is present in the Get<ControlBlock>Value response.

NOTE Generally, the namespace owner shall prefer extensions which do not imply that the subscriber shall be updated even if the functionality is not used.

K.3.5 Use case 34: Rename a service parameter not associated with a control block attribute

Fully compatible as the parameter name is not used in the mapping, only the parameter position is relevant in the service.

K.3.6 Use case 35: Removing / Deprecating Control Block Classes

Deprecation / removal of a control block class may be applied if the functionality is not needed anymore if a better approach has been defined to model the functionality. Table K.27 shows the impacts/recommendations related to the use case.

Table K.27 – Deprecating control block class

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> guaranteed.</p> <p><i>Backwards compatibility:</i> achievable. See K.1.5.3.</p>
Subscriber	<p><i>Forward compatibility:</i> not achievable if the deprecated control block class is associated to the subscribed publisher process. Otherwise guaranteed.</p> <p><i>Backwards compatibility:</i> achievable. See K.1.5.3</p>

After a period of deprecation, the Control Block Class can be removed from the standard and its use is only allowed between systems that supported them. The deprecation time is occurring to allow configurations to stop using the feature. Backward compatibility is not achievable and not required.

K.3.7 Use case 36: Using a stronger presence condition

Changing presence condition may be done for corrections. Table K.28 shows the impacts/recommendations related to the use case.

Table K.28 – Using a stronger presence condition

Concerned entity	Impact / recommendations
Client	<p><i>Forward compatibility:</i> achievable.</p> <p>CRu: client shall implement functionality associated to all control block attributes.</p> <p><i>Backwards compatibility:</i> achievable</p> <p>Requires a specific compatibility rule if control block attribute is mandatory.</p>
Subscriber	<p><i>Forward compatibility:</i> achievable.</p> <p>CRu: subscribers shall implement functionality associated to all control block attributes.</p> <p><i>Backwards compatibility:</i> achievable.</p> <p>Requires a specific compatibility rule if control block attribute is mandatory.</p>

Clients compliant with the current revision of the standard shall be aware that servers compliant to previous revisions of the standard may not expose the control block attribute and associated semantic / functionality applying the specific compatibility rule.

Bibliography

IEC 60050-101:1998, *International Electrotechnical Vocabulary – Part 1: Mathematics*

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

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
info@iec.ch
www.iec.ch