

INTERNATIONAL STANDARD

NORME INTERNATIONALE



AMENDMENT 1 AMENDEMENT 1

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

Réseaux et systèmes de communication pour l'automatisation des systèmes électriques – Partie 7-1: Structure de communication de base – Principes et modèles



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.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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 It also gives information on projects, replaced and withdrawn publications.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.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et définitions des publications IEC parues entre 2002 et 2015. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



AMENDMENT 1
AMENDEMENT 1

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

**Réseaux et systèmes de communication pour l'automatisation des systèmes
électriques –
Partie 7-1: Structure de communication de base – Principes et modèles**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 33.200

ISBN 978-2-8322-8481-0

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

FOREWORD

This amendment has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

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

FDIS	Report on voting
57/2201/FDIS	57/2221/RVD

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

The committee has decided that the contents of this amendment and the base publication will remain unchanged until the stability date indicated on the IEC website 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.

4 Abbreviated terms

Add the following new terms to the list of abbreviated terms:

MICS	Model implementation conformance statement
PICS	Protocol implementation conformance statement
SICS	SCL implementation conformance statement

5 Overview of the IEC 61850 series concepts

5.3 The information models of substation automation systems

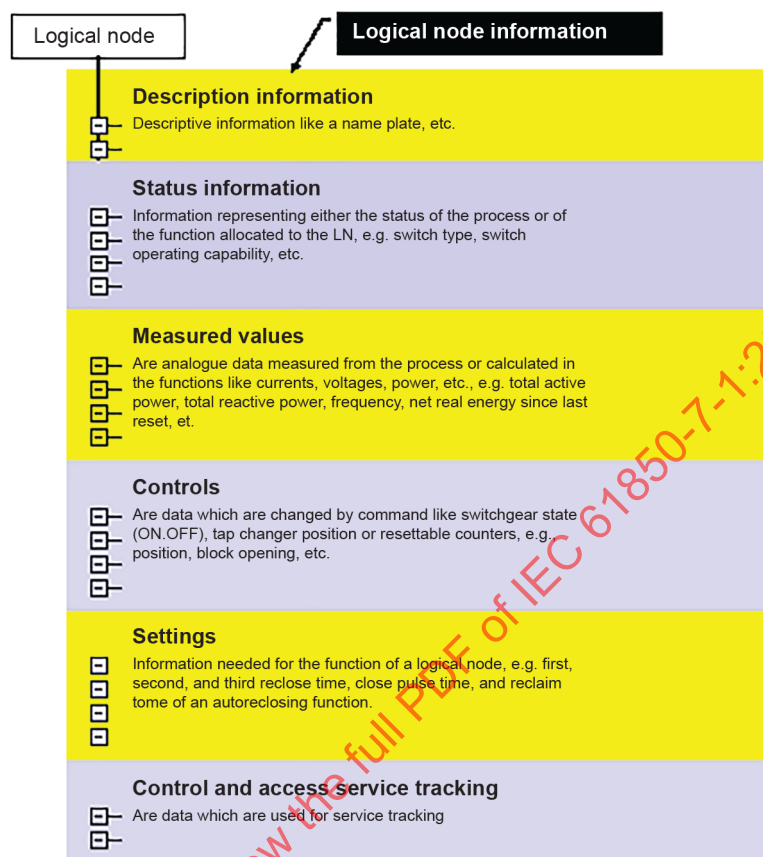
Add, at the end of the third paragraph below Figure 3, the following new footnote:

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

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

Figure 4 – Logical node information categories

Replace existing Figure 4 with the following new figure:

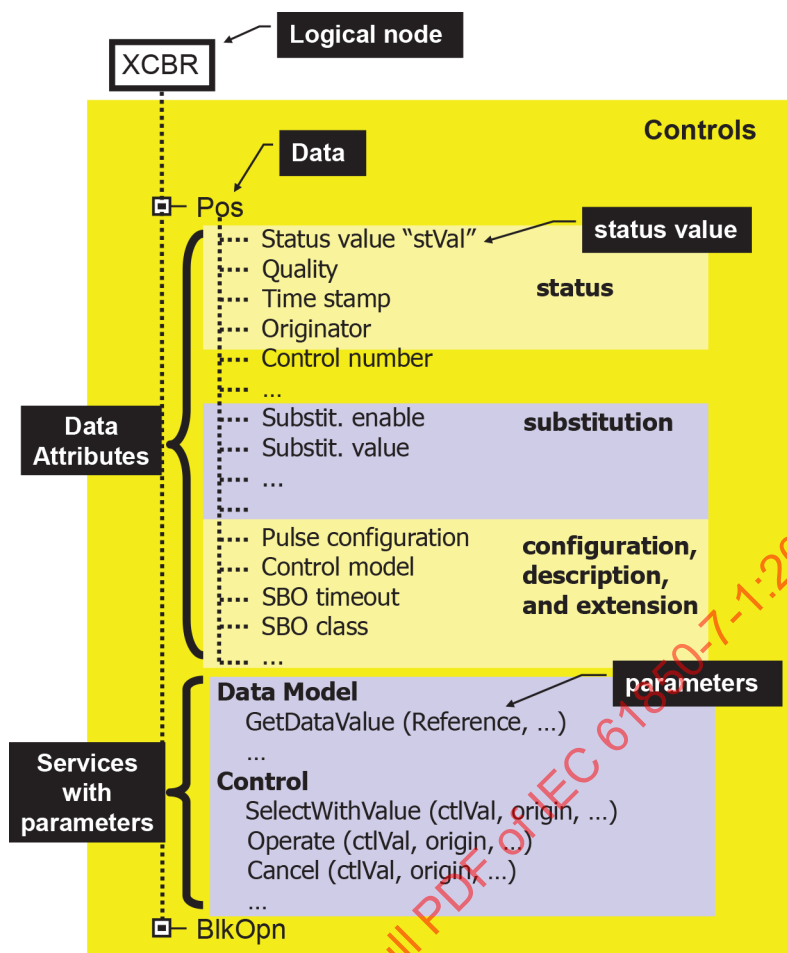


IEC

5.5 The semantic is attached to data

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

Replace existing Figure 6 with the following new figure:



IEC

Replace the first table of Subclause 5.5 with the following new table:

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

Below this table, replace the last item of the dashed list with the following new item:

- 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.

After the dashed list, replace the first paragraph and the following table with the following new paragraph:

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.

6 Modelling approach of the IEC 61850 series

6.2 Creating information models by stepwise composition

Table 2 – Logical node class XCBR (conceptual)

Replace existing Table 2 with the following new table:

Descriptions
External equipment name plate
Name plate of the logical node
Controls
Switch position (see below for details)
Block opening
Block closing
Charger motor enabled
Control authority at station level. Switches between station and higher level.
...
Status information
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
...

Replace the existing table below Figure 11 with the following new table:

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

6.4.2 Output model

6.4.2.1 Control model concept

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

Replace existing Figure 14 with the following new figure:

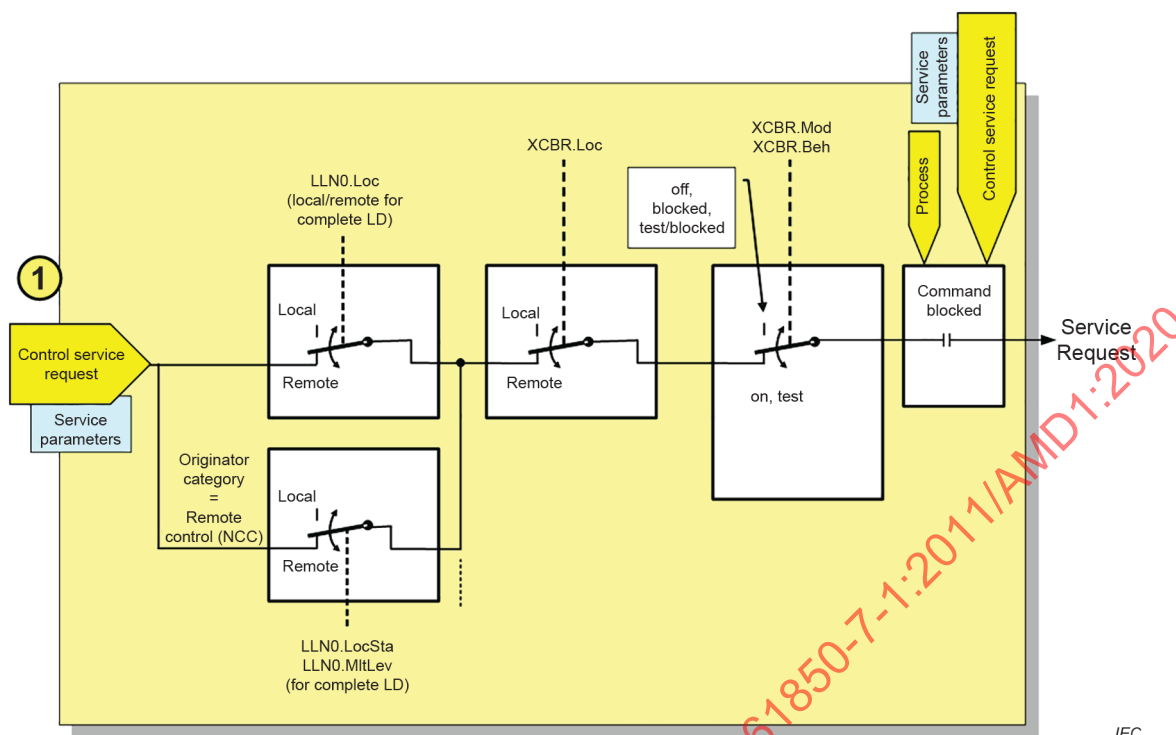


Figure 14 – Output model (step 1) (conceptual)

6.4.2.2 GSE and SMV model concept

Add the following new text after the paragraph below Figure 16:

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.

6.4.2.4 Setting data and setting group control block

Table 3 – Excerpt of integer status setting

Replace existing Table 3 by the following new table:

ING class					
Attribute name	Attribute type	FC	TrgOp	Value/value range	PresCond
DataAttribute for configuration, description and extension					
setting					
setVal	INT32	SP	dchg	The value of the status setting.	AtMostOne
setVal	INT32	SG, SE		The value of the status setting.	AtMostOne
configuration, description and extension					
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
...	...	...			...

6.4.3 Input model

6.4.3.2 Data attribute value processing, monitoring and event detection

Replace the last sentence of the second paragraph with the following new sentence:

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.

Replace Figure 19 and the subsequent paragraph with the following new figure and text:

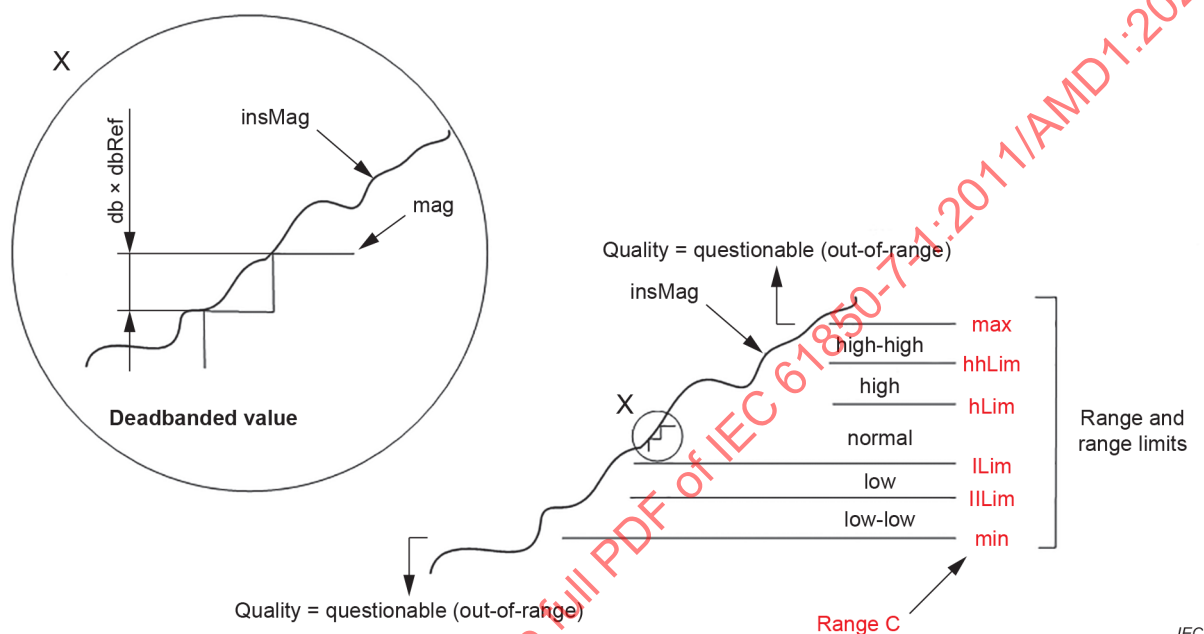
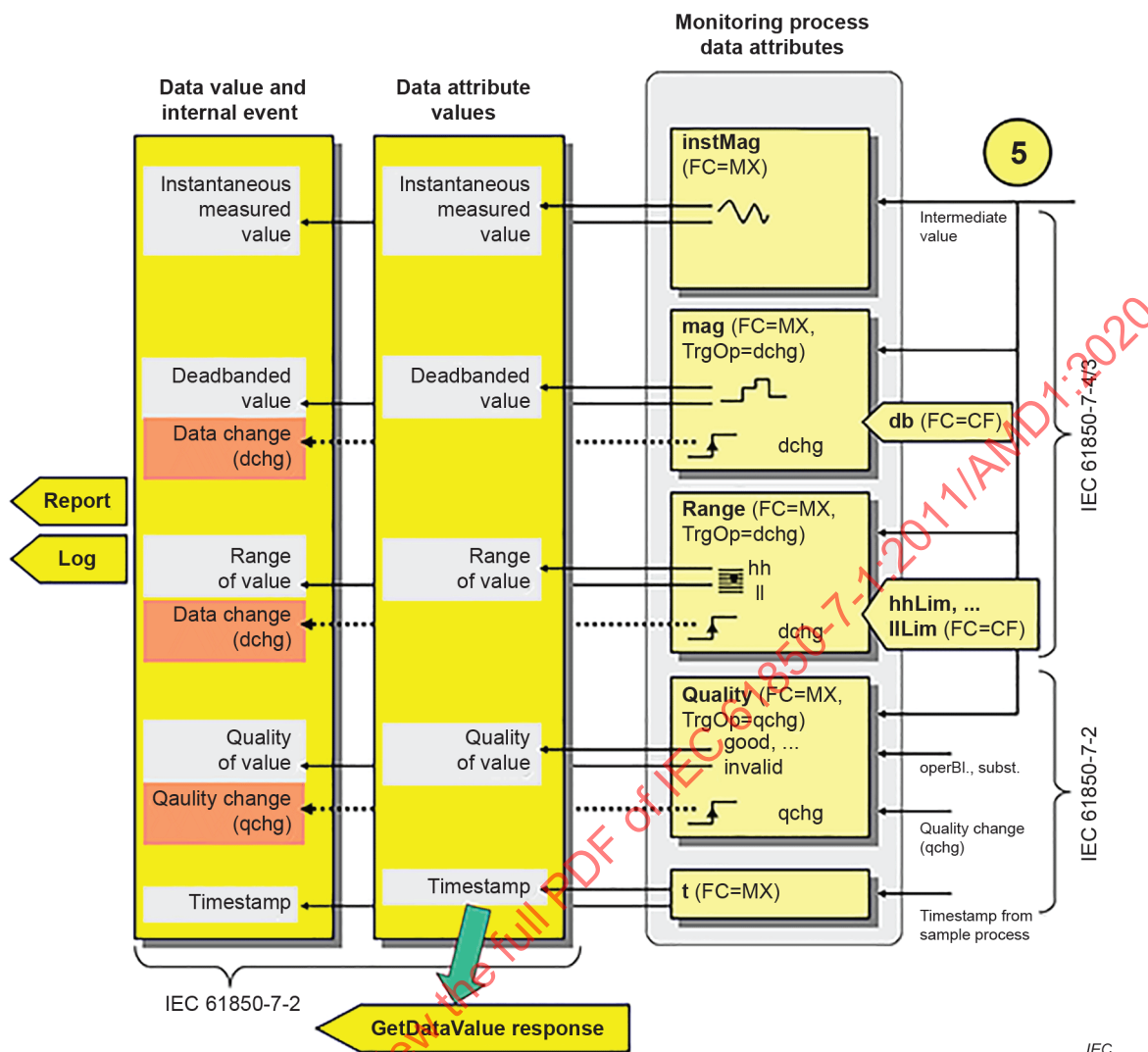


Figure 19 – Range and deadbanded value (conceptual)

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).

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

Replace existing Figure 20 with the following new figure:



Remove the fourth paragraph below Figure 20.

6.4.3.3.2 Data reporting

Replace existing Table 4 with the following new table:

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	Server	Engineering stations

Replace the existing paragraph before Figure 23 with the following new text:

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.

Before the last sentence of the first paragraph below Figure 23 add the following new sentence.

The buffering is independent from the reporting being enabled or not.

6.4.3.4 Peer-to-peer data value publishing

Replace the last paragraph of 6.4.3.4 by the new following new text:

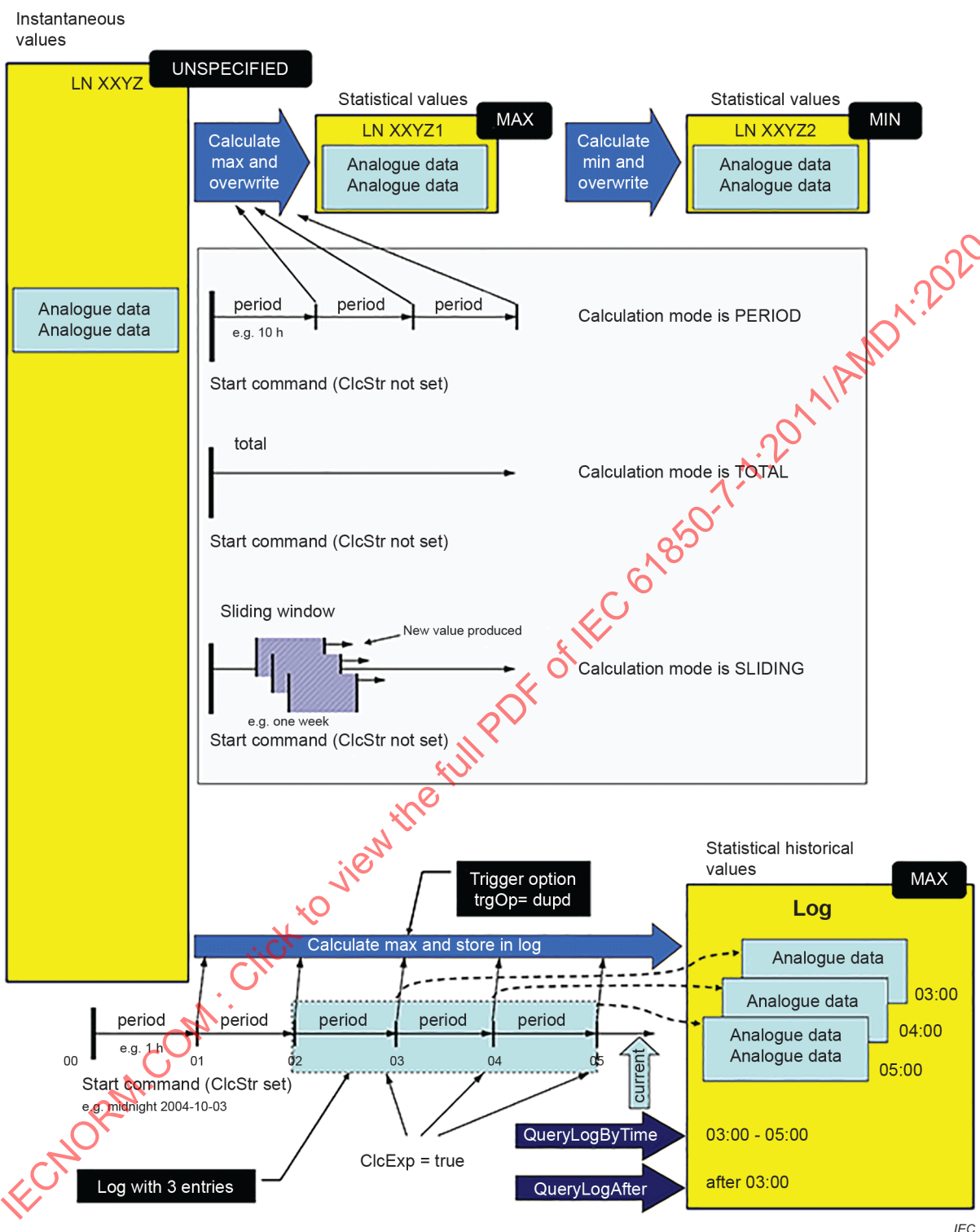
Which event triggers the publishing of values as well as how often and how fast the values are to be published is a matter of the requirements of the function being implemented.

6.4.4 Model for statistical and historical statistical data

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

Replace existing Figure 28 with the following new figure:

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020



IEC

Replace the penultimate paragraph before Figure 29 with the following new text:

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.

Remove the last paragraph of Subclause 6.4.4.

6.4.5 Model for system functions

6.4.5.1 General

Remove the last two sentences of Subclause 6.4.5.1.

6.4.5.5 GOOSE and SMV supervision

Add the following new text at the end of Subclause 6.4.5.5:

The normative Annex G describes the configuration flow with regards to these supervision logical nodes.

7 Application view

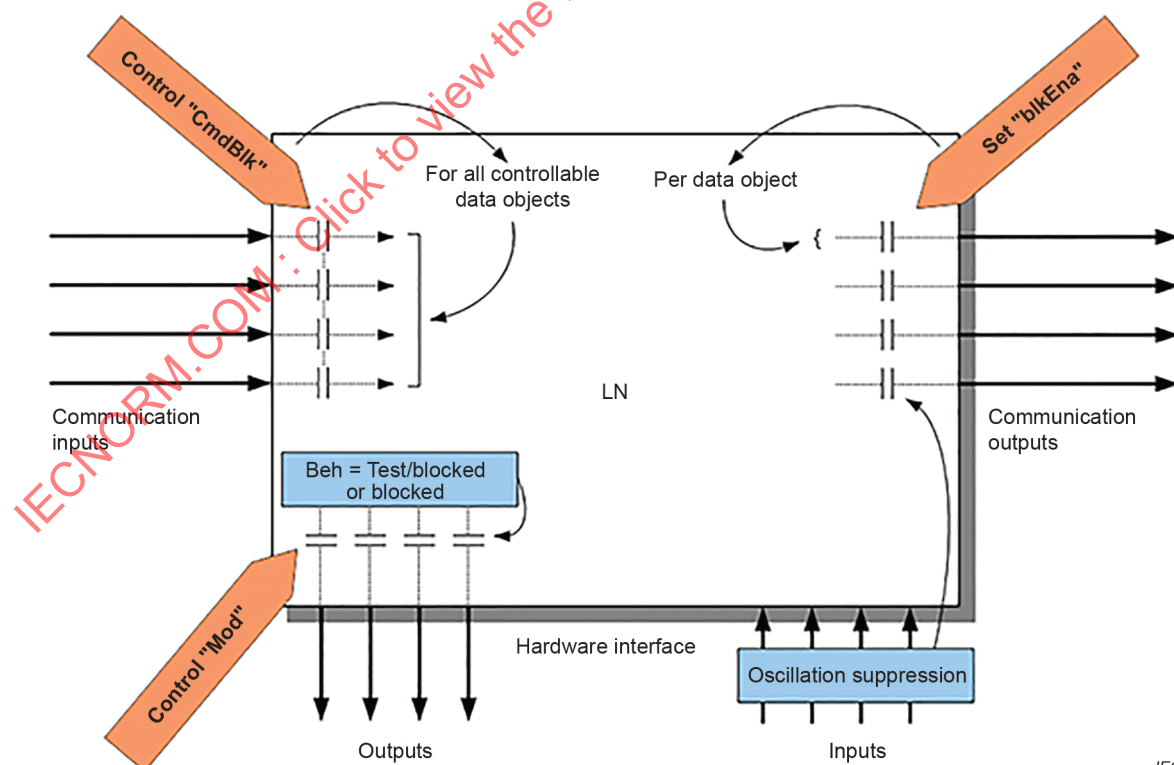
7.3 Mode and behaviour of a logical node

Replace “on-blocked” by “blocked” in the first paragraph below Figure 37.

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

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

Replace existing Figure 39 by the following new figure:



IEC

7.7.2 Blocking incoming commands

Add, at the end of the last paragraph of Subclause 7.7.2, the following new text:

or CmdBlk itself.

7.7.3 Blocking process outputs

Replace the existing text of Subclause 7.7.3 by the following new text:

The data Mod (Mode) is used to put an LN in different modes. The mode test/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 oscillating inputs

Replace the title of 7.7.4 with the following new text:

Blocking the communication of status outputs updates

Replace the first sentence of 7.7.4 with the following new text:

An operator, or an automatic function, may block the update of the value of communication outputs (e.g. updates related to oscillating inputs).

7.8 Data used for testing

7.8.2 Multicast signals used for simulation

Replace the third paragraph of 7.8.2 with the following new text:

To allow the IED1 to process the simulated GOOSE 1 message instead of the actual GOOSE message, the data Sim.stVal in the LN LPHD1 shall be set to TRUE. The process of the simulated message only takes effect upon receiving the first simulated message. This will remain as such until Sim.stVal is set back to 'FALSE'. The processing of the simulated message is made on a per GOOSE/SMV basis, i.e. the other signals, GOOSE 2 and GOOSE 3, are normally processed by the IED1. Note that the switching between normal signals and simulated signals 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 Figure 80 and Figure 81 that describes the different states involved and illustrates the supervision of GOOSE/SV subscription with regards to the active subscription state resp. the subscription to simulated stream. The affected Data Objects are defined in the Logical Nodes LGOS and LSVS in IEC 61850-7-4.

Add, at the end of the third paragraph of 7.8.2, the following new Figures 80 and 81:

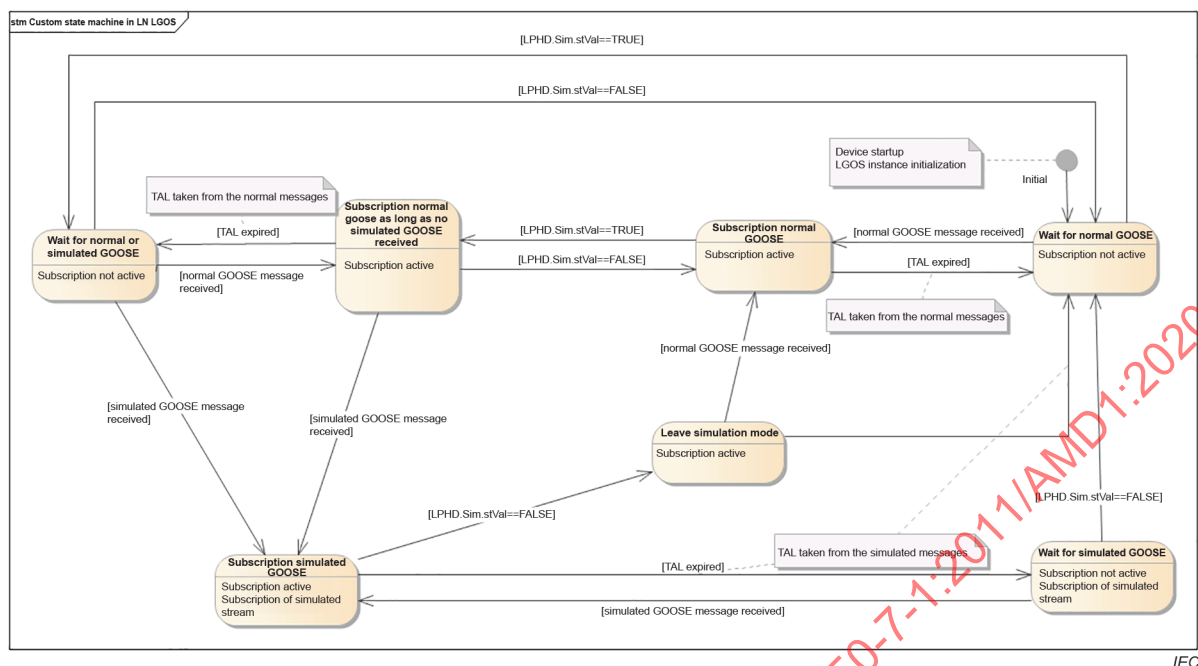


Figure 80 – GOOSE subscription supervision state machine

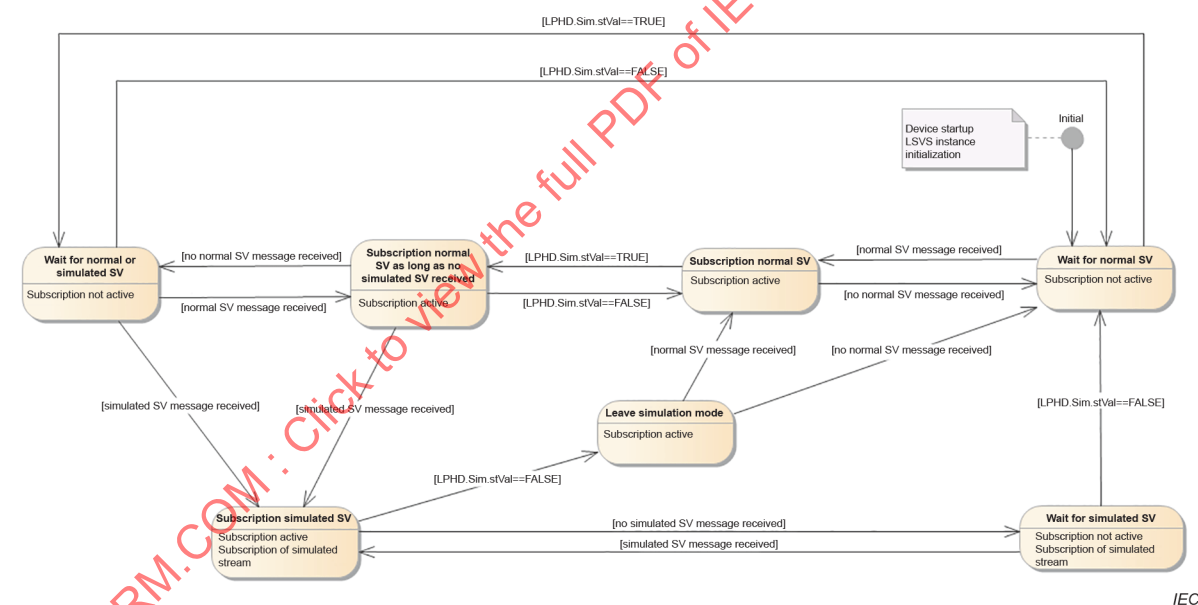
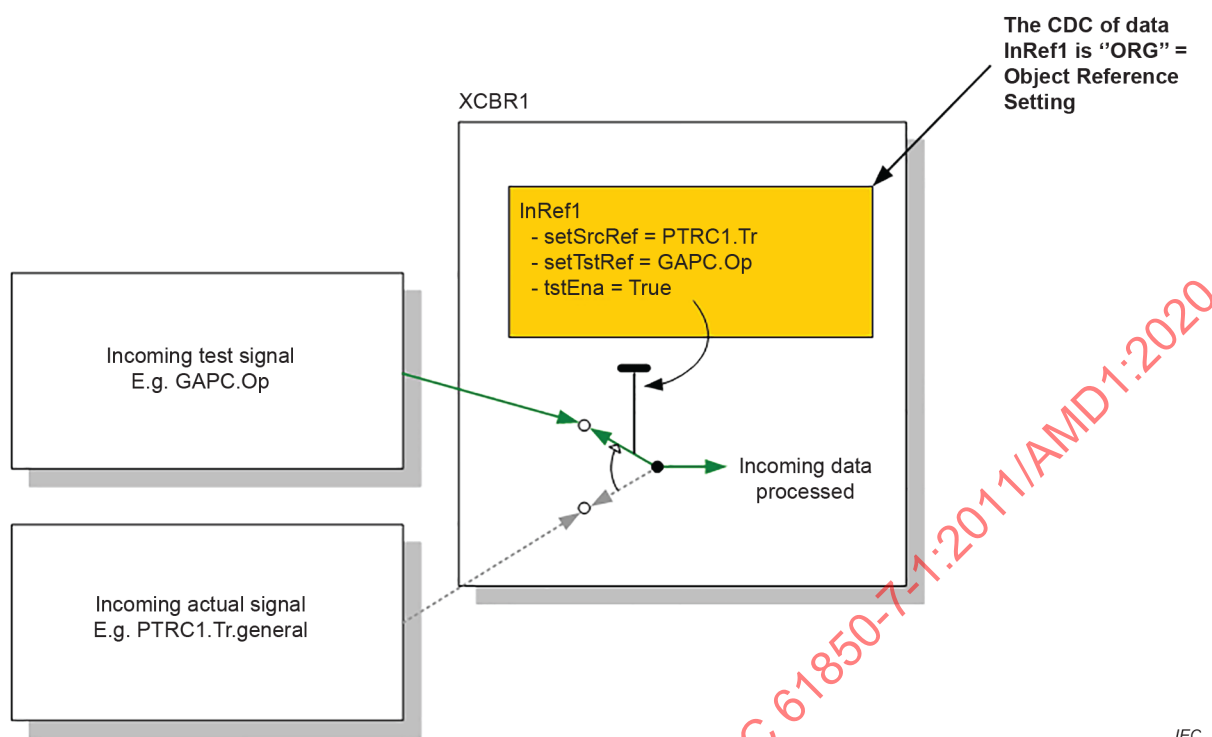


Figure 81 – SV subscription supervision state machine

7.8.3 Input signals used for testing

Figure 41 – Example of input signals used for testing

Replace existing Figure 41 with the following new figure:



IEC

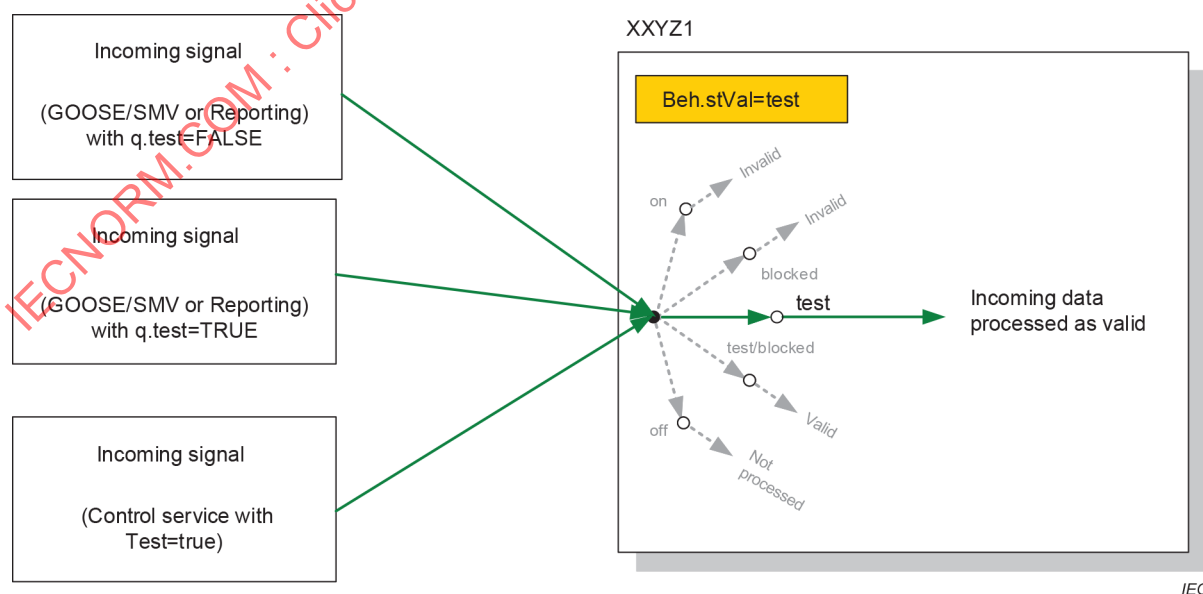
Add the following new text at the end of the last paragraph of 7.8.3:

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

Replace existing Figure 42 by the following new figure:



IEC

Replace the last paragraph of Subclause 7.8.4 with the following new text:

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

8 Device view

8.2 Second modelling step – logical device model

8.2.1 The logical device concepts

Replace the second paragraph below Figure 44 with the following new text:

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).

8.2.3 Gateways and proxies

Replace the existing text of Subclause 8.2.3, including Figure 47, with the following new text and figures:

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., IED1LD1). 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_GTW1LD1, 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.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

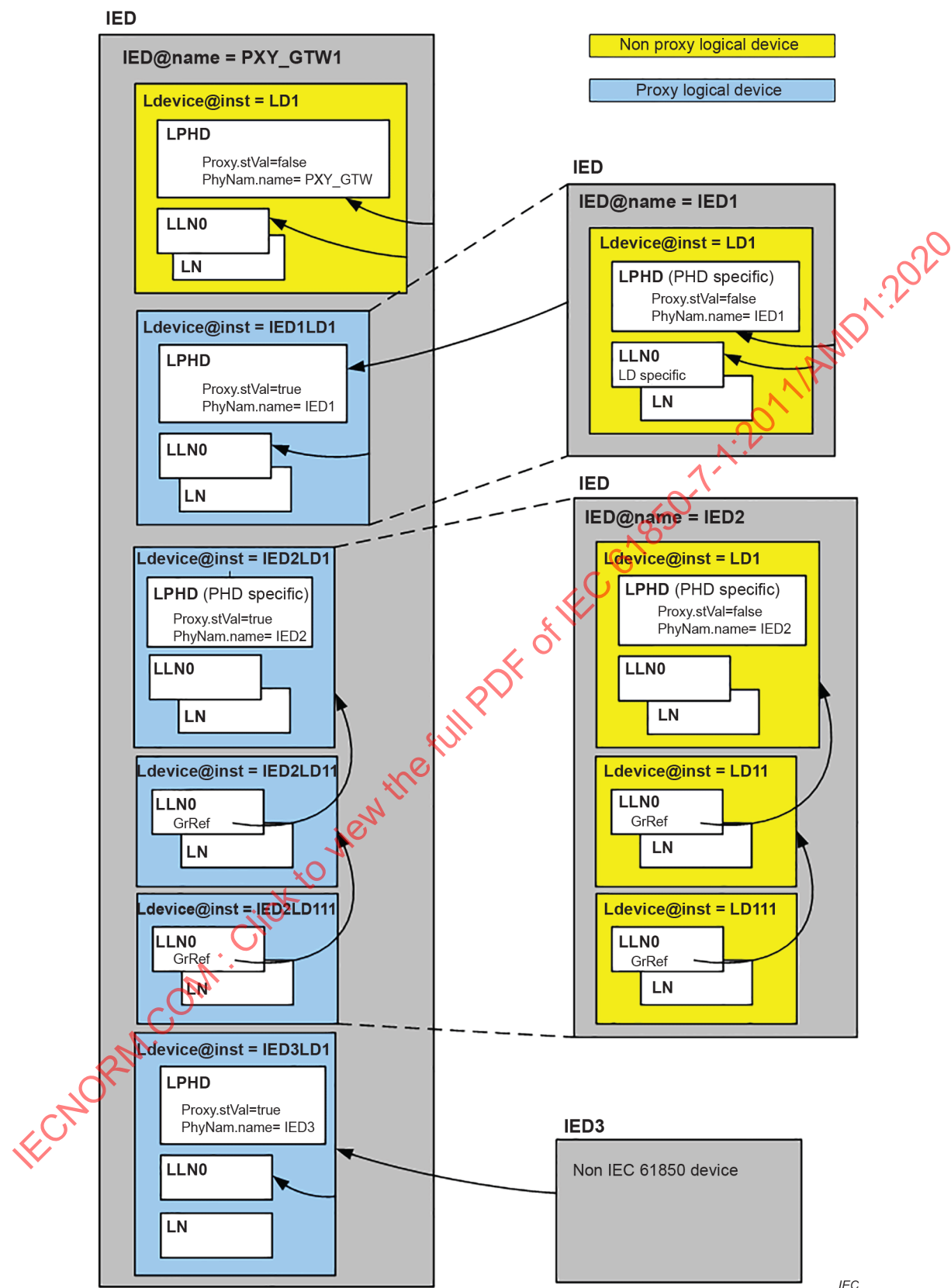


Figure 47 – Logical devices in proxies or gateways

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.

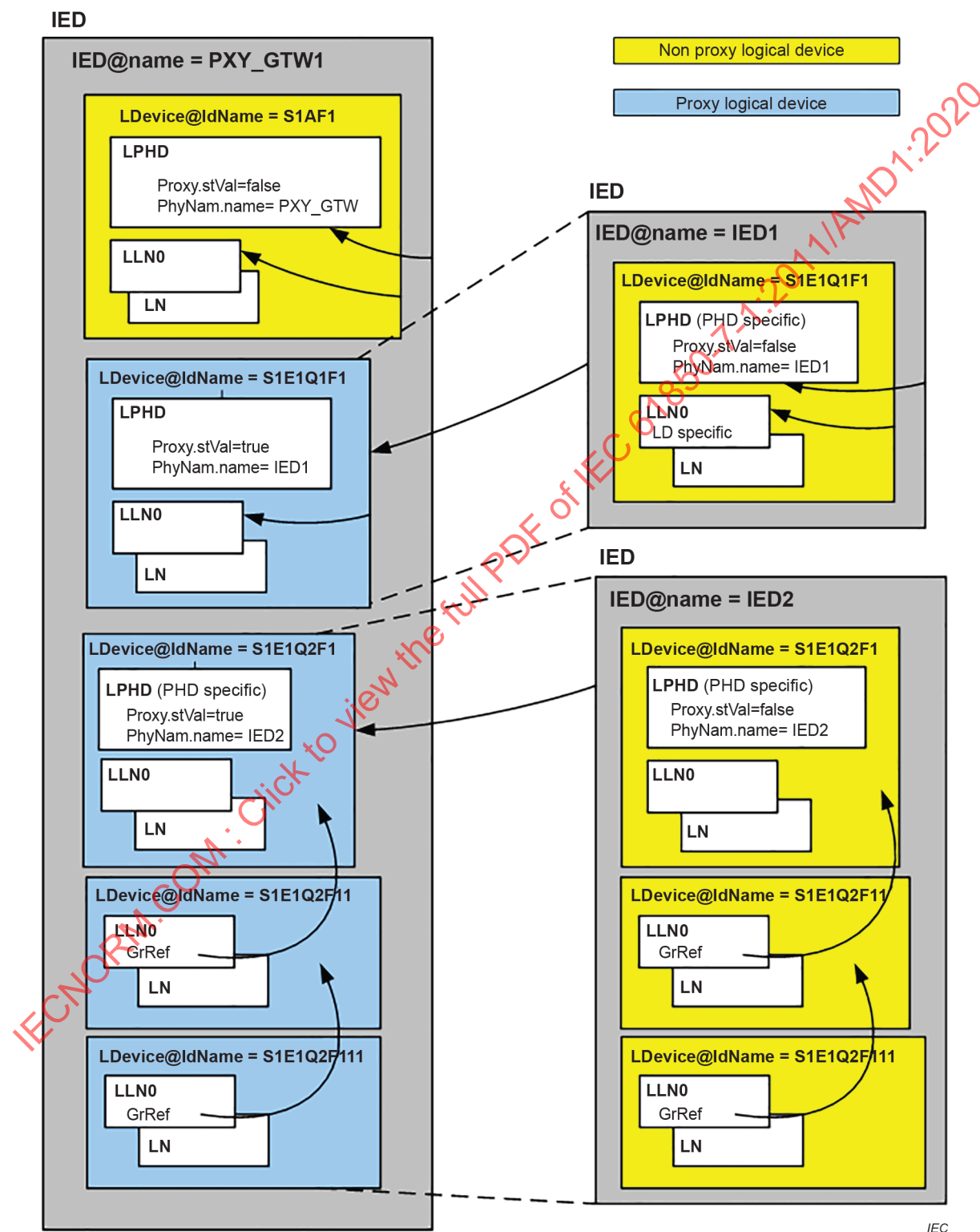
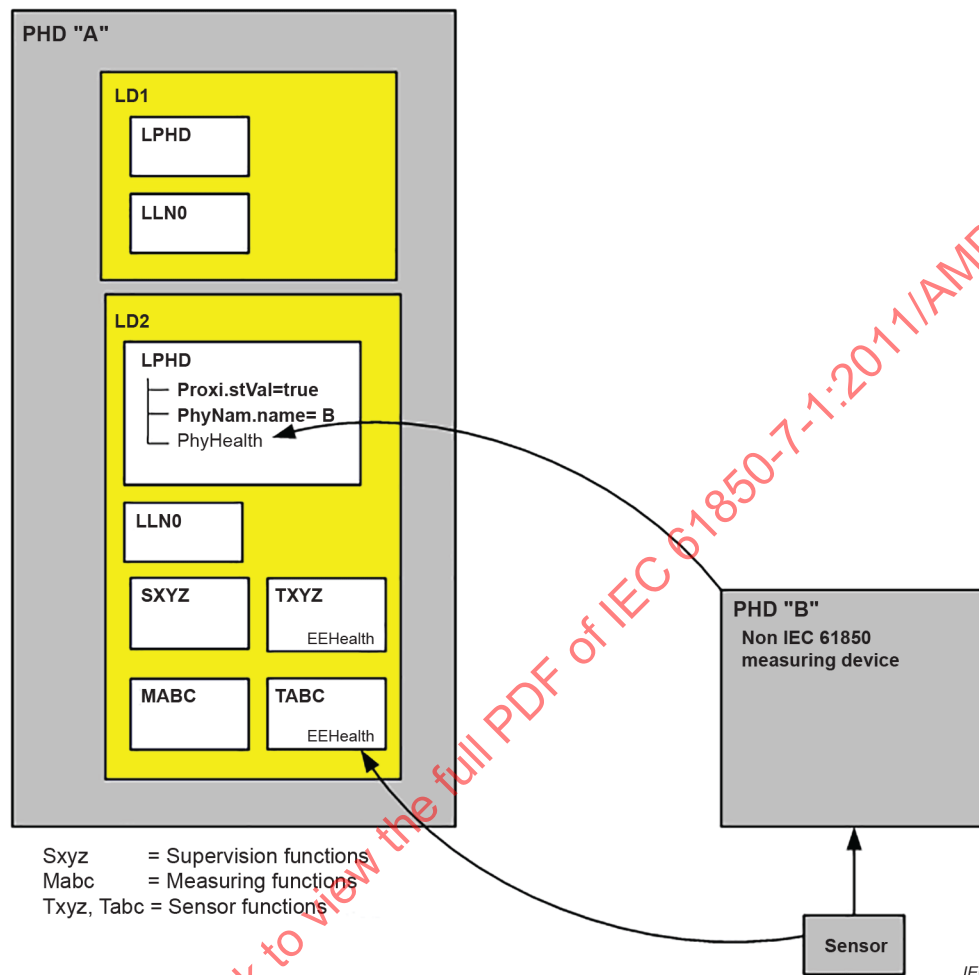


Figure 79 – Logical devices in proxies or gateways (functional naming)

8.2.4 Logical devices for monitoring external device health

Figure 48 – Logical devices for monitoring external device health

Replace existing Figure 48 with the following new figure:



8.2.5 Logical devices management hierarchy

In the second paragraph of Subclause 8.2.5, replace "object reference setting group" with "object reference setting".

Replace the third and fourth items in the dashed list with the following new text:

- 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,

9 Communication view

9.2 The service models of the IEC 61850 series

Table 5 – ACSI models and services

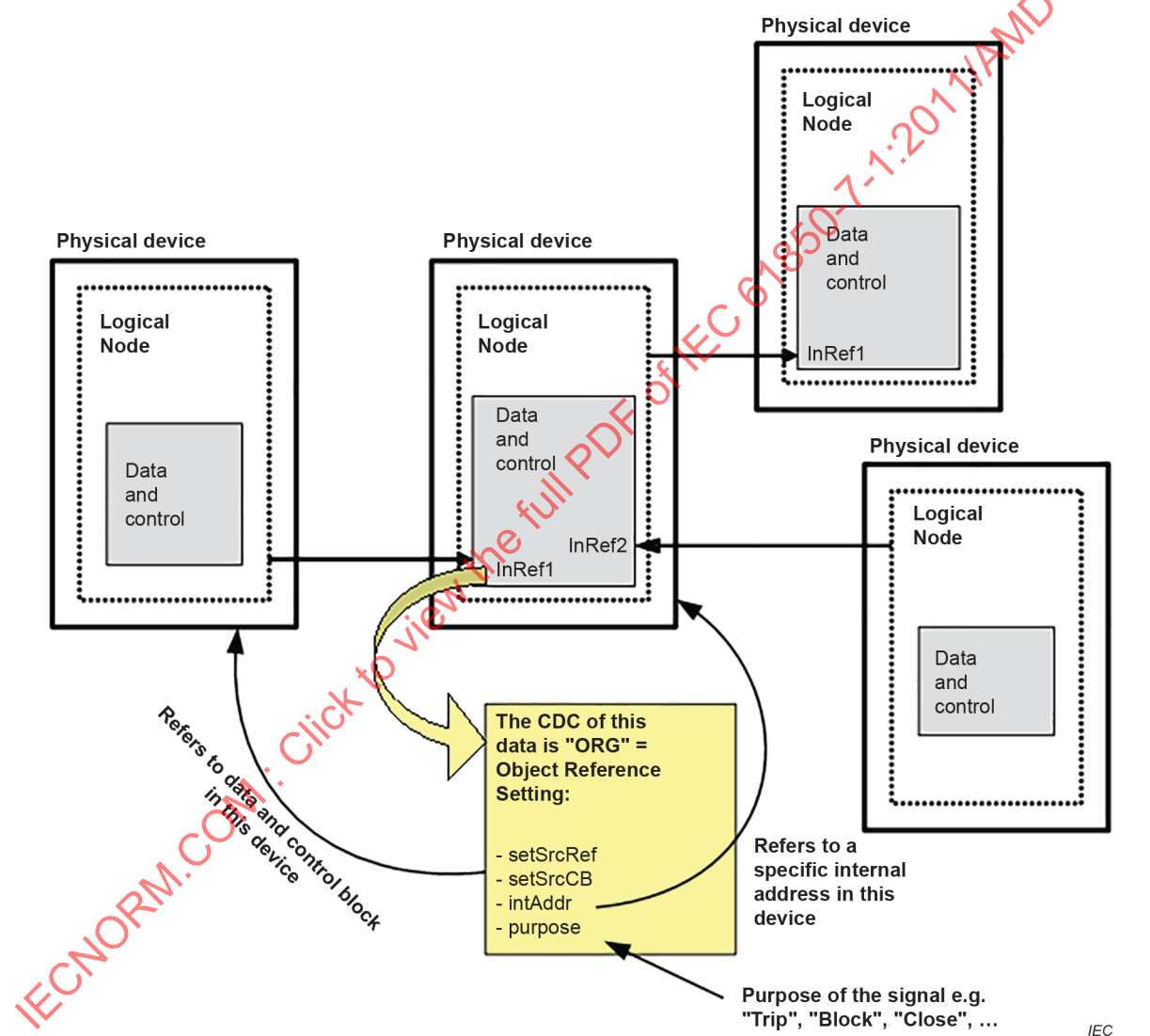
Replace, in Table 5, the row "Transmission of sampled values" with the following new row.

Transmission of sampled values	Fast and cyclic transfer of samples, for example, of instrument transformers.	Multicast SVC: SendMSVMessage GetMSVCBValues SetMSVCBValues GetMsvReference GetMSVElementNumber
--------------------------------	---	--

9.6 Logical nodes communicate with logical nodes

Figure 60 – Logical nodes communicate with logical nodes

Replace existing Figure 60 with the following new figure:



Replace the last paragraph of Subclause 9.6 with the following new text:

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).

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

11.2 Example 1 – Logical node and data object class

Replace the first paragraph of Subclause 11.2 with the following new text:

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).

Replace existing Table 6 by the following new table:

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

Table 6 – Logical node circuit breaker

Logical Node: Circuit breaker		Name: XCBR	
Data object Class	Data object Name	Common Data Class (CDC)	PresConds/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.	BlkOpn	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.	BlkCls	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
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.	LocKey	SPS – Single point status	O/F

Logical Node: Circuit breaker		Name: XCBR	
Data object Class	Data object Name	Common Data Class (CDC)	PresCond nds/ds
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

Replace existing Table 7 by the following new table:

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

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).	

Replace existing Figure 64 by the following new figure:

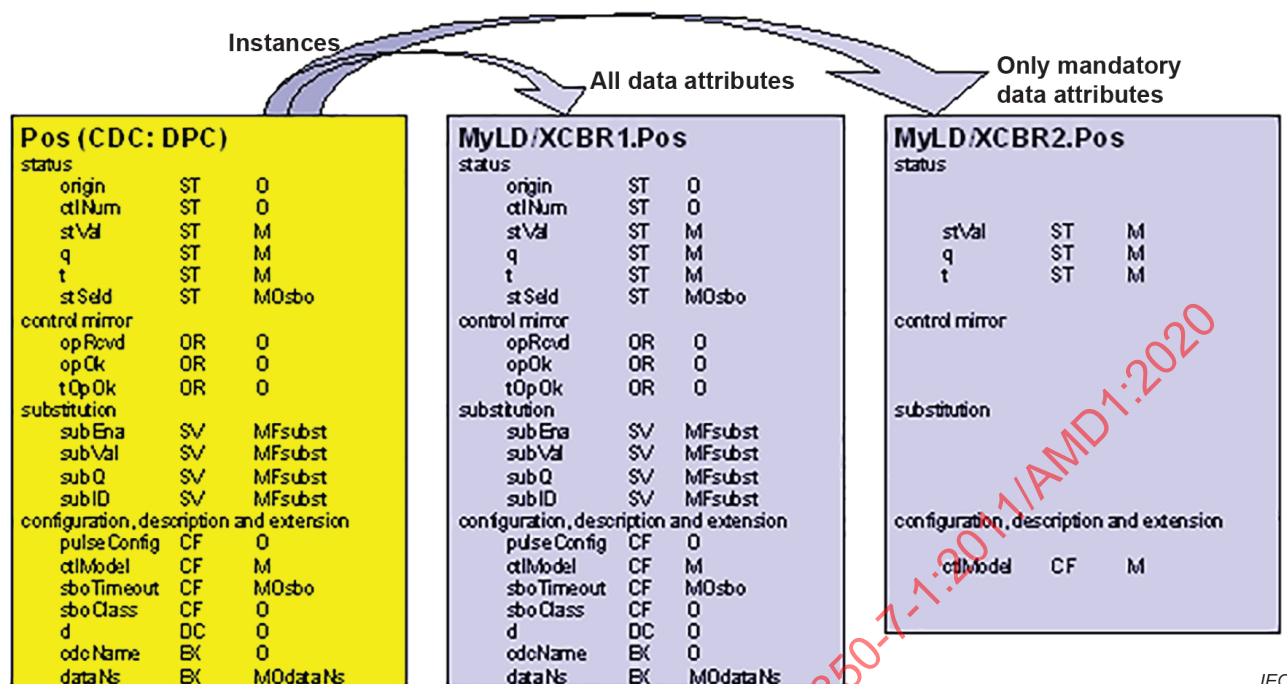


Figure 64 – Instances of a Data object class (conceptual)

Replace the second paragraph following Figure 64 with the following new text:

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

Replace the existing table following the second paragraph after Figure 64 by the following new table:

Abbreviation	Condition
M	Attribute is mandatory
O	Attribute is optional
MOsbo	Element is mandatory if declared control model supports 'sbo-with-normal-security' or 'sbo-with-enhanced-security', otherwise optional and value is of no impact.
MFsubst	Element is mandatory if substitution is supported (for substitution, see IEC 61850-7-2), otherwise forbidden.
MOenhanced	Element is mandatory if declared control model supports 'direct-with-enhanced-security' or 'sbo-with-enhanced-security', otherwise optional and value is of no impact.
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.
...	...

12 Formal specification method

12.2 Class modelling

12.2.1 Overview

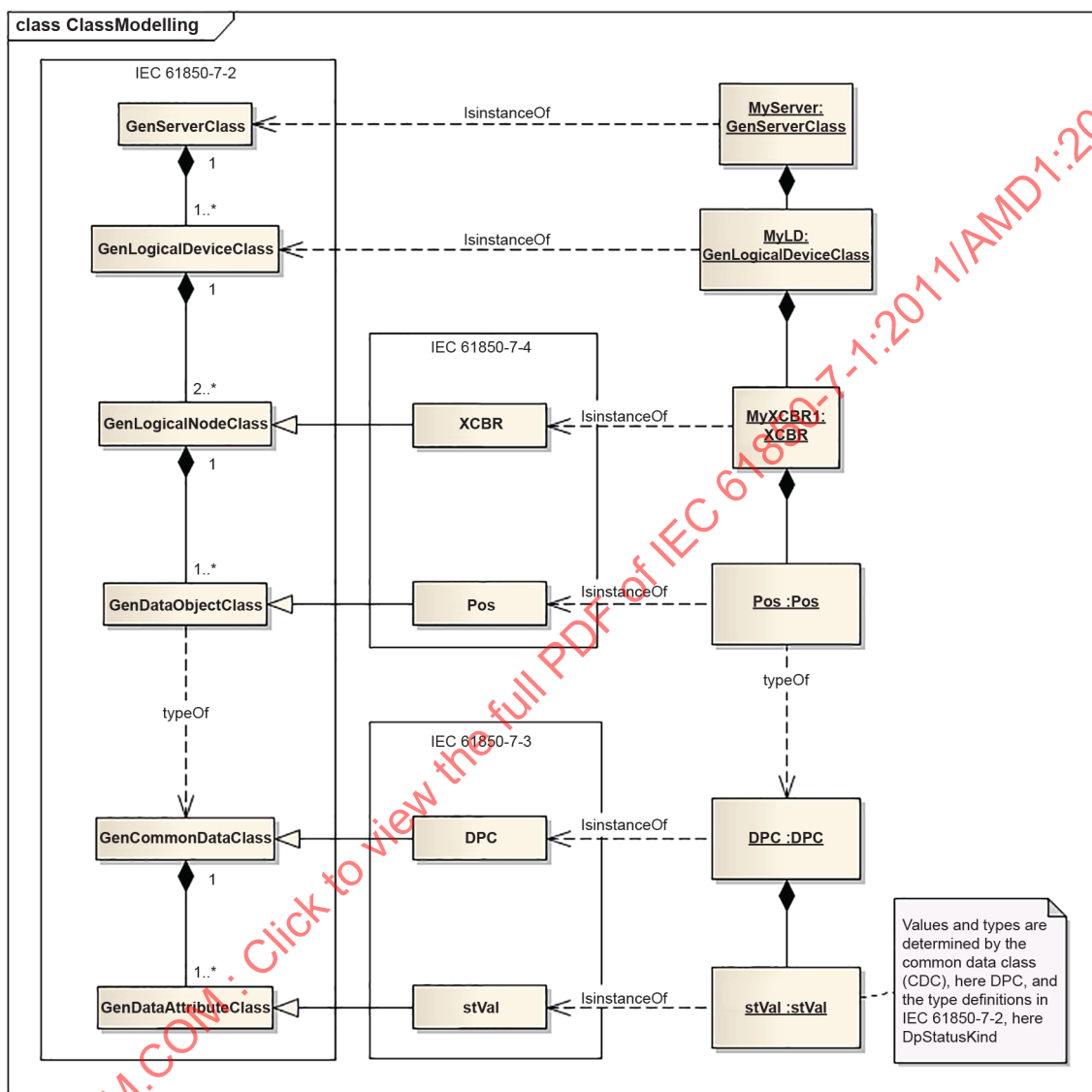
Replace the second paragraph of Subclause 12.2.1 with the following new text:

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

Replace existing Figure 66 with the following new figure:



IEC

12.2.2 Common data class

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

Replace existing Table 9 with the following new table:

cdcid = 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

Replace the existing second and third paragraphs of Subclause 12.2.2 with the following new text:

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

Replace existing Table 10 with the following new table:

atId = 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

Delete the paragraph following Table 10 and add the following new Table 16 after Table 10:

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
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

Add the following new text after Table 16:

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 – Basic status information template (excerpt)

Replace the existing Table 11 with the following new table:

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.

Replace the existing text between Tables 11 and 12 with the following new text:

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

Replace existing Table 12 with the following new table:

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.

Replace the last paragraph of Subclause 12.2.2 with the following new text:

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 – GenLogicalNodeClass definition

Replace existing Table 13 with the following new table:

GenLogicalNodeClass		
Attribute name	Attribute type	Explanation
LNName	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/LNName .
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 namespace 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 namespace 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 namespace IEC 61850-7-4.		

Replace the existing ninth and tenth paragraphs of Subclause 12.2.3 with the following new text:

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.

12.3 Service parameters tables

Replace the first table of Subclause 12.3 with the following new table:

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		

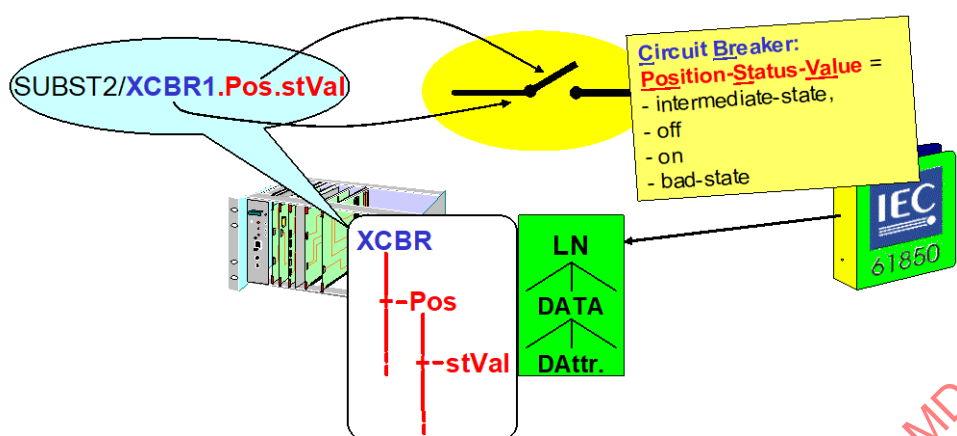
13 Namespaces

Replace the existing text, figures and tables of Clause 13 with the following new text, figures and tables:

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 or other class namespace of the IEC 61850 series, or related to the IEC 61850 series 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 namespace, 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 namespace 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 namespace information to allow the unambiguous interpretation of the semantic of the instance. The instances of classes are marked to identify the namespace.

13.2 Namespaces defined in the IEC 61850 series or related to the IEC 61850 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.

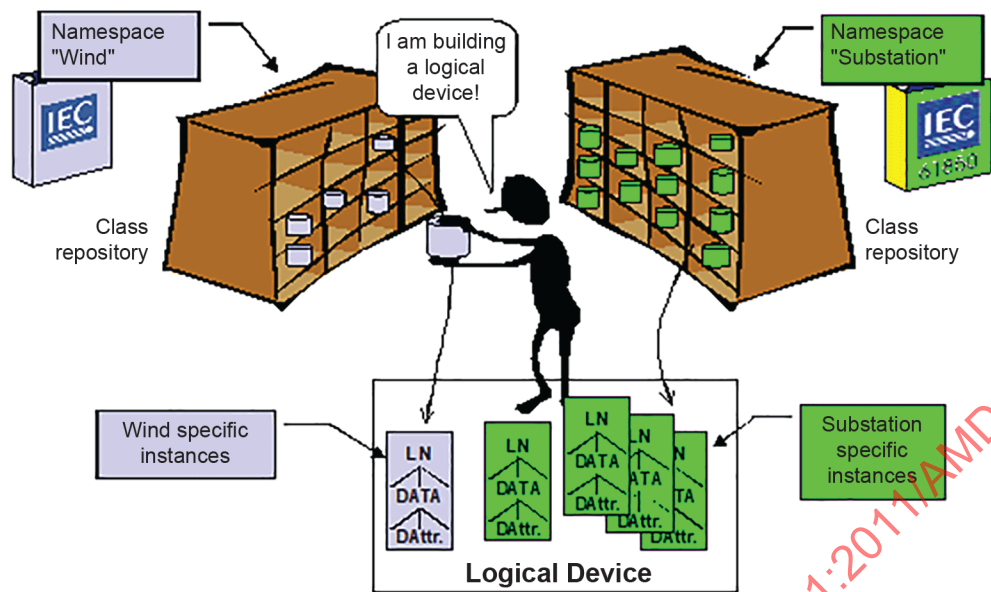


Figure 73 – Namespace as class repository

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 namespace is represented by the attribute “logical device namespace”:

IdNs = IEC 61850-7-4:2003

In this example, the namespace “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 namespace could be understood as the prime namespace; even if there is just one name. The attribute IdNs (logical device namespace) 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 namespace (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.

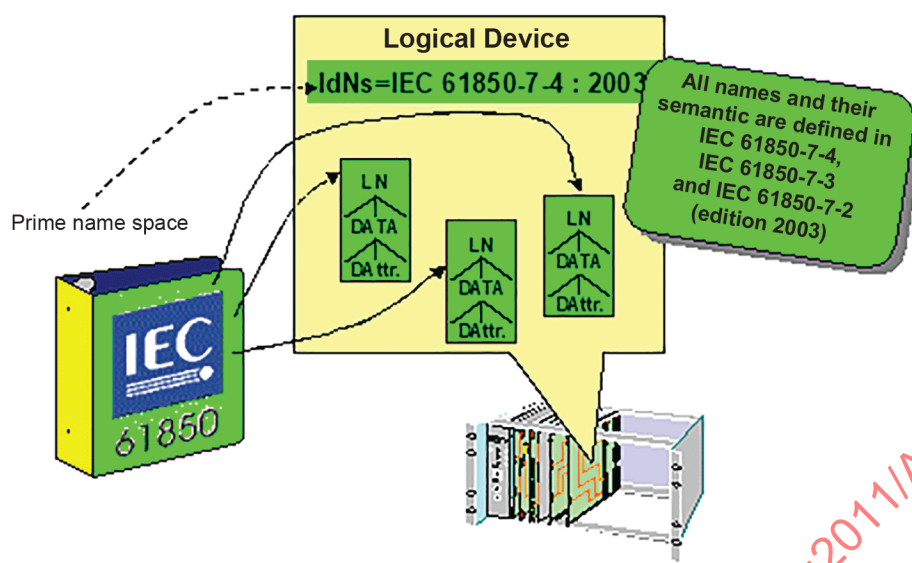


Figure 74 – All instances derived from classes in a single namespace

When extensions (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 specific 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 namespace (Vestas). Since the majority of instances are derived from IEC 61850-7-4, the logical device namespace IdNs is still IEC 61850-7-4 (prime namespace).

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

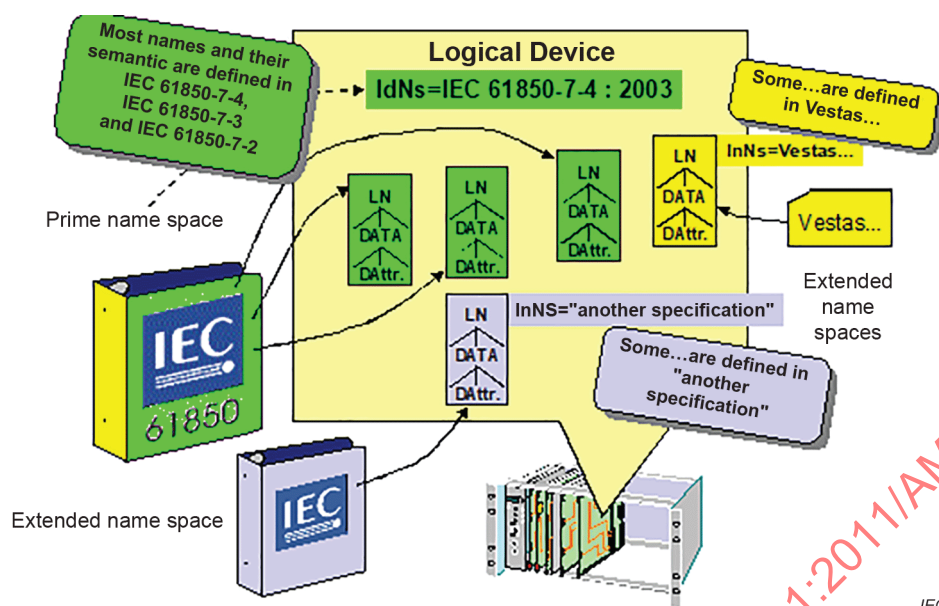


Figure 75 – Instances derived from multiple namespaces

The logical devices namespace could be “IEC 61850-7-4:2003” or any other domain namespace (as defined below) in which a logical device is defined and used depending on the context and which “incudes” at minimum a version of IEC 61850-7-4. 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 namespace

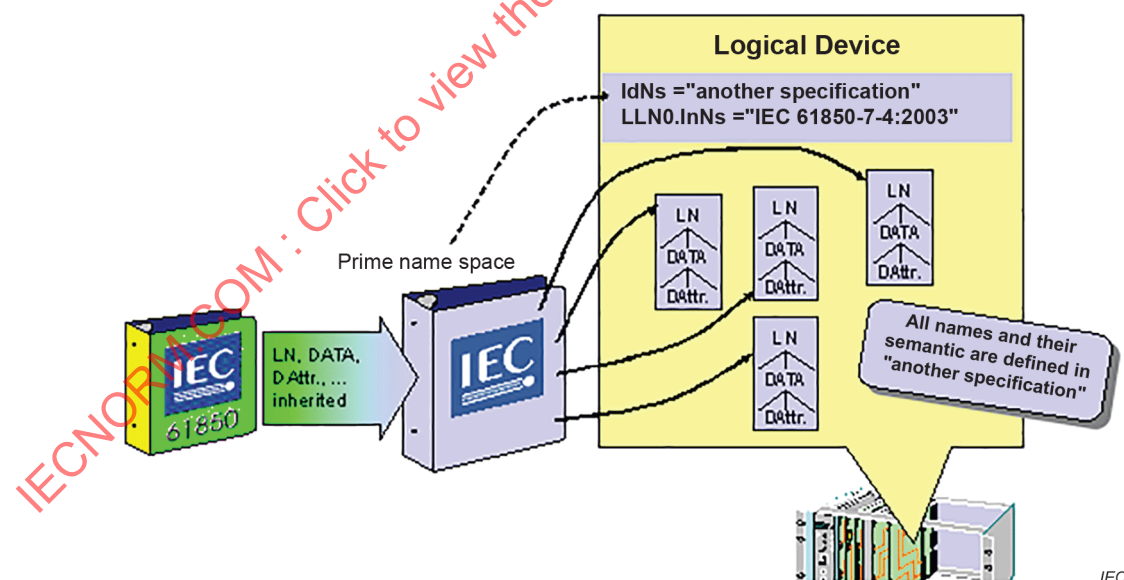


Figure 76 – Inherited namespaces

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 namespace. 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 mostly 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. Applicable for product standard and transitional namespaces.

NOTE "Depends on", "Includes", "Extends" are machine code references and shall not be translated.

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 namespace. Entities may own several namespaces.

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

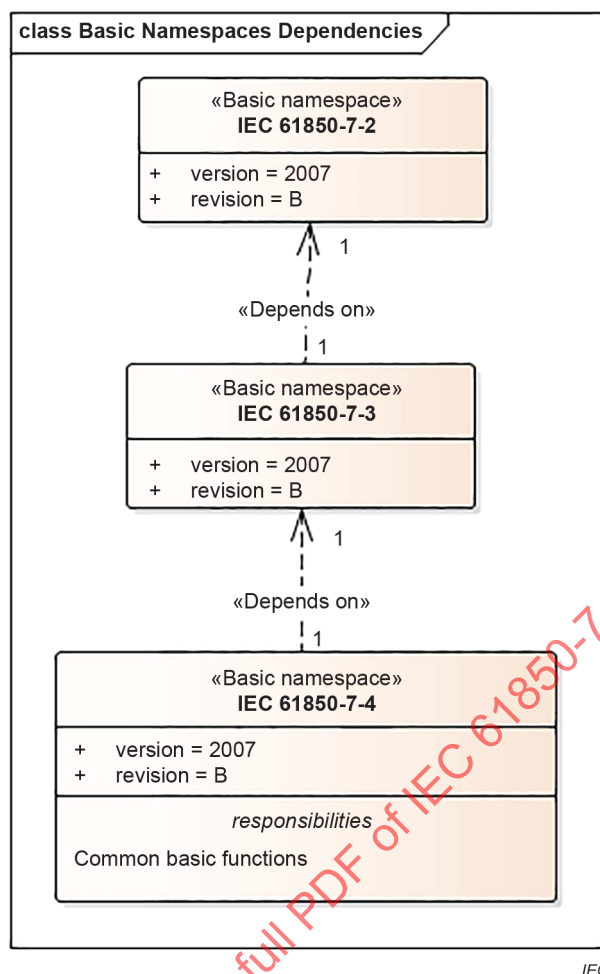


Figure 77 – Basic namespaces dependencies

The basic namespaces covered by the IEC 61850 series are:

- IEC 61850-7-2 which specifies:
 - Basic types, domain types and service types and enumerations,
 - 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.

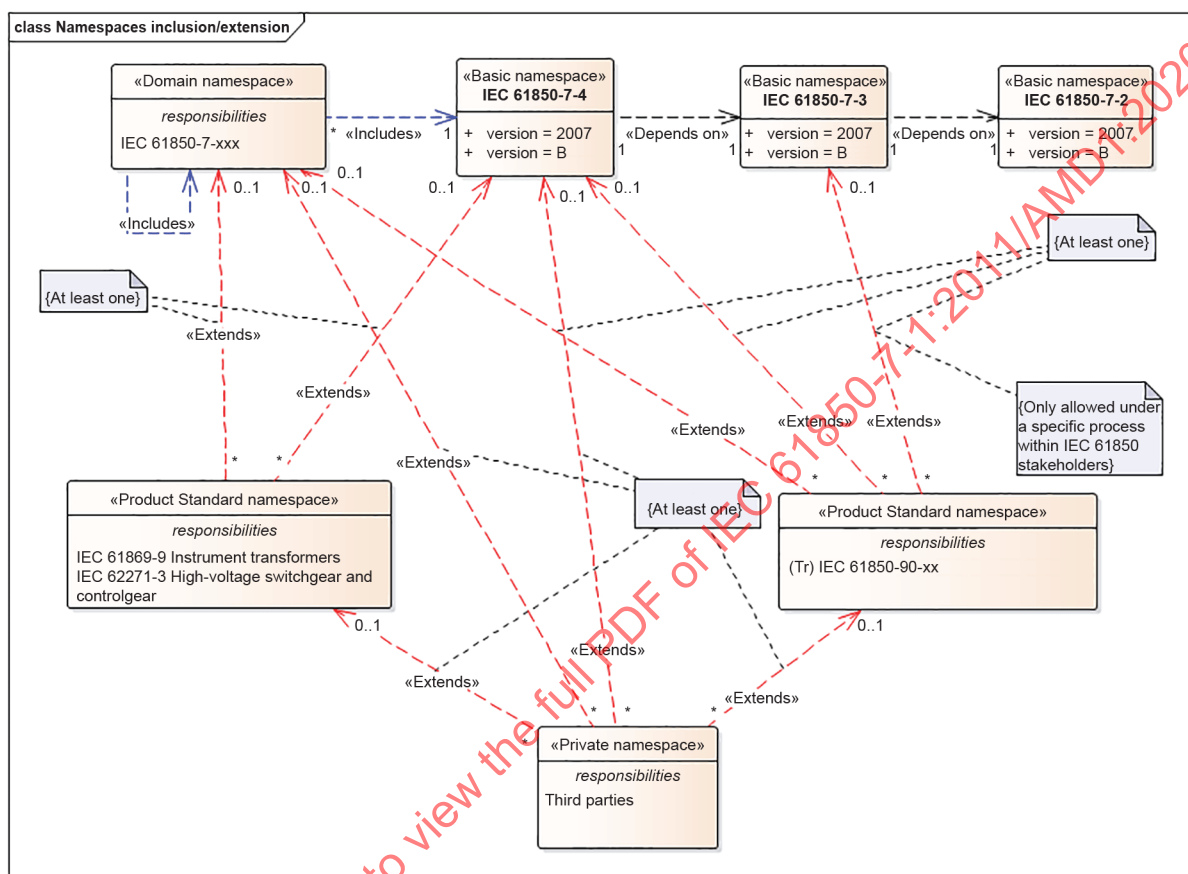
The namespaces for domain applications covered by the IEC 61850 series are:

- 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.



IEC

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 and versioning of namespaces

Standardized namespace names shall be structured as follows:

NamespaceID:VersionRevision

Where

- NamespaceID is the identification of the namespace. NamespaceID shall refer to a technical specification (paper specification, schema (XML, database), UML description, etc.),
 - For normative documents:
 - NamespaceID = IEC {IEC reference number}, e.g. IEC 61850-7-4
 - For Technical Reports:
 - NamespaceID = (Tr)IEC {IEC reference number}, e.g. (Tr)IEC 61850-90-4. In addition, the following accompanying text shall be added to the definition of the namespace name:
'The content of this namespace shall be considered as a transitional namespace. This means that the content may be moved to an IS status with some changes, thus making possible backward incompatibilities with existing implementations.'
- Transitional namespaces shall be deprecated after they have been implemented in a domain namespace (refer to 13.6). Note that the document may be kept as place holder for the use case descriptions,
- Version refers to an arbitrary year number which should be as close as possible to the year of first publication of the concerned edition, and shall be kept unchanged for amendment and corrigenda of the same edition. 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, ...

For versioning purpose, a release number is added to the definition of namespaces with the purpose of capturing "micro" evolutions of namespaces, especially to take into account the needed fixes of interoperability issues. The release parameter is an incremental number between revisions (integer starting from 0 – the default value 0 is optionally mentioned). Release number shall not be used as part of the standardized namespace name used as reference to namespaces (as defined in 13.5.1 and when expressing namespace dependencies).

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

13.5 Attributes for references to namespaces

13.5.1 General

The following five attributes that reference namespaces are defined:

- Attribute logical device namespace (ldNs) shall reference the domain application namespace used as the prime technical specification used for the whole logical device.
- Attribute logical node namespace (lnNs) of the LLN0 shall reference the basic namespace version/revision used for the whole logical device.
- Attribute logical node namespace (lnNs) shall reference the logical node namespace of a single instance of a compatible logical node class.
- Attribute data namespace (dataNs) shall reference the data namespace of a single instance of a compatible data object class.
- Attribute common data class namespace (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) Description	PresCond
DataAttribute for configuration, description and extension					
...	...	...	...	...	...
ldNs	VisString255	EX		Logical device namespace, for example "IEC 61850-7-4:2007B".	MFln0
lnNs	VisString255	EX		Logical node namespace.	MOlnNs

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) Description	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 namespace ldNs shall be used to indicate the domain application namespace used as the prime namespace for the whole logical device, e.g. IEC 61850-7-4, IEC 61850-7-410, IEC 61850-7-420, etc.

The attribute IdNs shall be a DataAttribute of the name plate NamPlt of the logical node class LLN0. The attribute IdNs 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 IdNs shall be available in every instance of the logical node class LLN0.

The ObjectReference for the DataAttribute IdNs shall be:

LDName/LLN0.NamPlt.IdNs

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

The DataAttribute logical node namespace LLN0.NamPlt.InNs shall be used to indicate the used basic namespace for a logical device. The version/revision of this basic namespace dictates the basic namespace version/revision that is implemented in the device. Examples can be found in Clauses J.9 and I.7.

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 NamPlt of a logical node LLN0. The attribute InNs shall be as defined in the common data class LPL (logical node name plate) of the versioned / revisioned IEC 61850-7-3 included by the basic namespace.

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 the versioned / revisioned IEC 61850-7-3 included by the basic namespace.

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^{1,2}
- the private Namespace if the logical node has been defined outside the standardized Namespace.

¹ The change of a presence condition in a product standard is considered as an extension.

² Standardized extension of an enumeration is considered as an extension.

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. NamPIt, Health, Mod)3. For all other data object classes, the attribute shall be available in an LN instance if its value differs from the lnNs value. The attribute is optional in case the dataNs and lnNs 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.

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 "issues" 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 issue must be based on the latest edition/version.

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

Replace the existing text, figures and tables of Clause 14 with the following new text, figures and tables:

14.1 General

This clause specifies rules for:

- Class version control.

³ DomainLN is defined in IEC 61850-7-4.

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

The description of the rules 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 transitional namespaces.
- Extensions made within basic and domain 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 78.

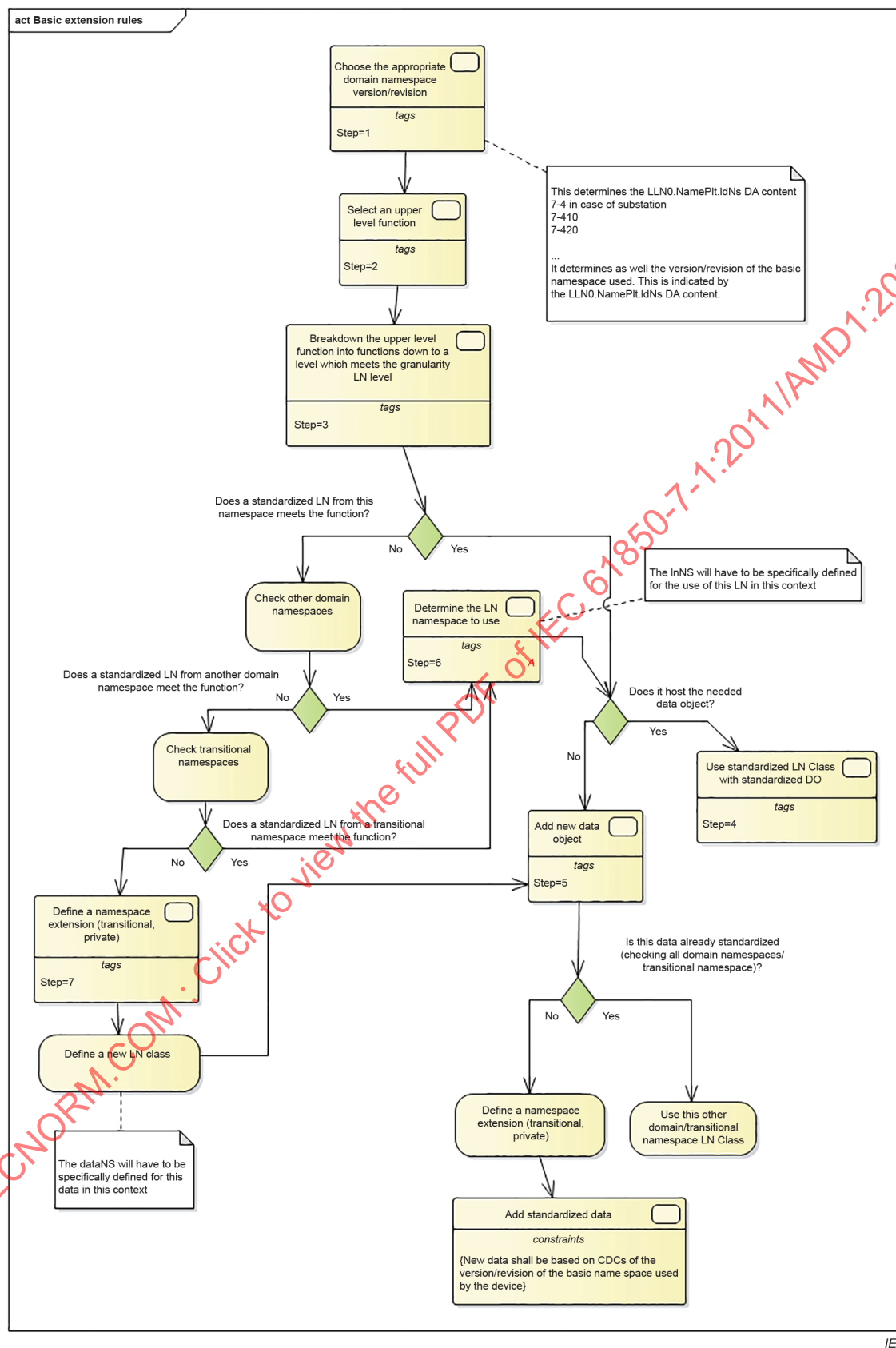


Figure 78 – Basic extension rules diagram

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.NamPit.IdNs data attribute. It will dictate as well what version/revision of the basic namespaces are intended to be implemented in the device. This basic namespaces 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 definitions⁴) 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.

⁴ 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.InNs⁵.

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 fulfil 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 shall 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 enumeration types and enumeration-based data objects and attributes

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

14.3 Extensions made 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 namespace name 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.
- d) Subsequent standardized data object class definitions (forward definitions) may be used. In such a case (see Clause J.4):

⁵ Refer to IEC 61850-7-4 for a definition of the DomainLN class.

- The semantic and type shall be the one of the standardized data object class.
 - In case of an LLN0 class extension, dataNs shall be mandatory and be the one of the standardized data object class.
 - For other compatible LN classes, the subsequent definition of the LN class shall be used (new version/revision of InNs)
- 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 J.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 matches 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 namespace. As a consequence, private extension of an existing CDC shall be prohibited.

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 does not 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 functions⁶**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 NamPlt (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 NamPlt (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 NamPlt (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 NamPlt (name plate).

⁶ The examples supplied in this clause are for informative purposes 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, 1,2,19,25),
- 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.In Ns
A for “Automatic Control”	D oor	E ntrance	C ontrol	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.In Ns
Z for “Further equipment”	F ire	P rotection	T ransformer	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

Add, after Clause 14, the following new Clause 15:

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. If for any reason this is not possible, general requirements and recommendations regarding compatibility issues are provided in Annex K.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

Annex A (informative)

Overview of logical nodes and data

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

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

Replace the existing text of Subclause A.1.3 with the following new text:

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.

Table A.1 – Excerpt of data classes for measurands

Delete Table A.1.

Annex B (informative)

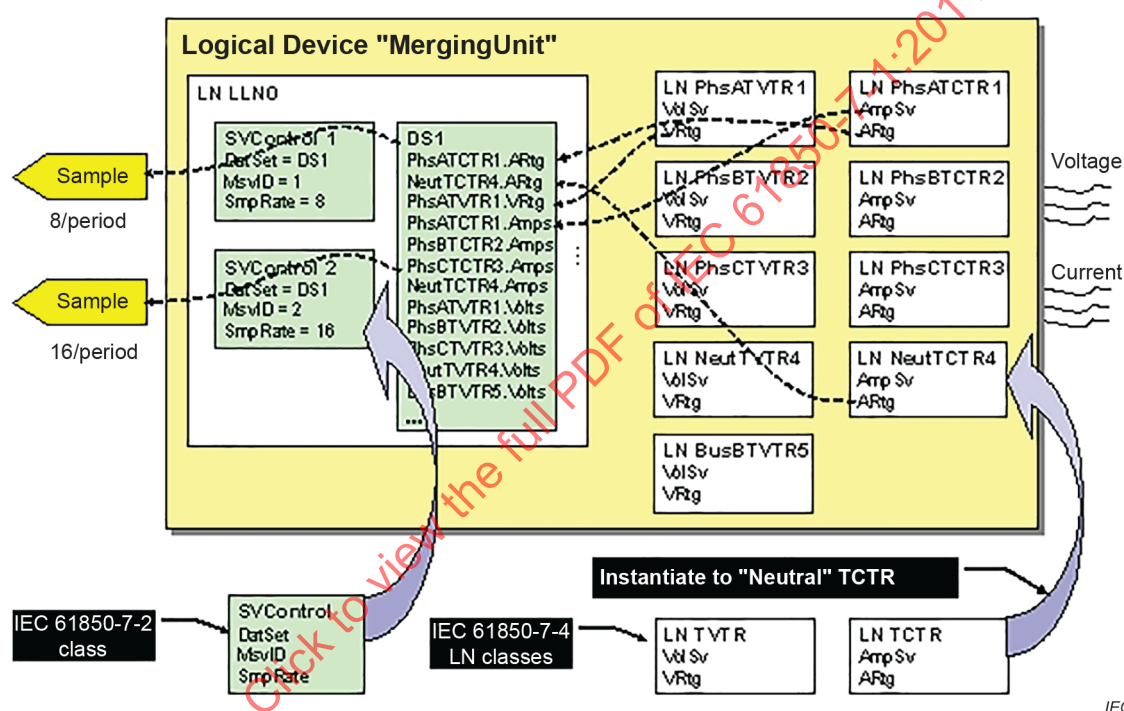
Allocation of data to logical nodes

Replace the existing first paragraph below Figure B.1 with the following new text:

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.

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

Replace existing Figure B.3 with the following new figure:



Replace the existing two paragraphs following Figure B.3 with the following new text:

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,

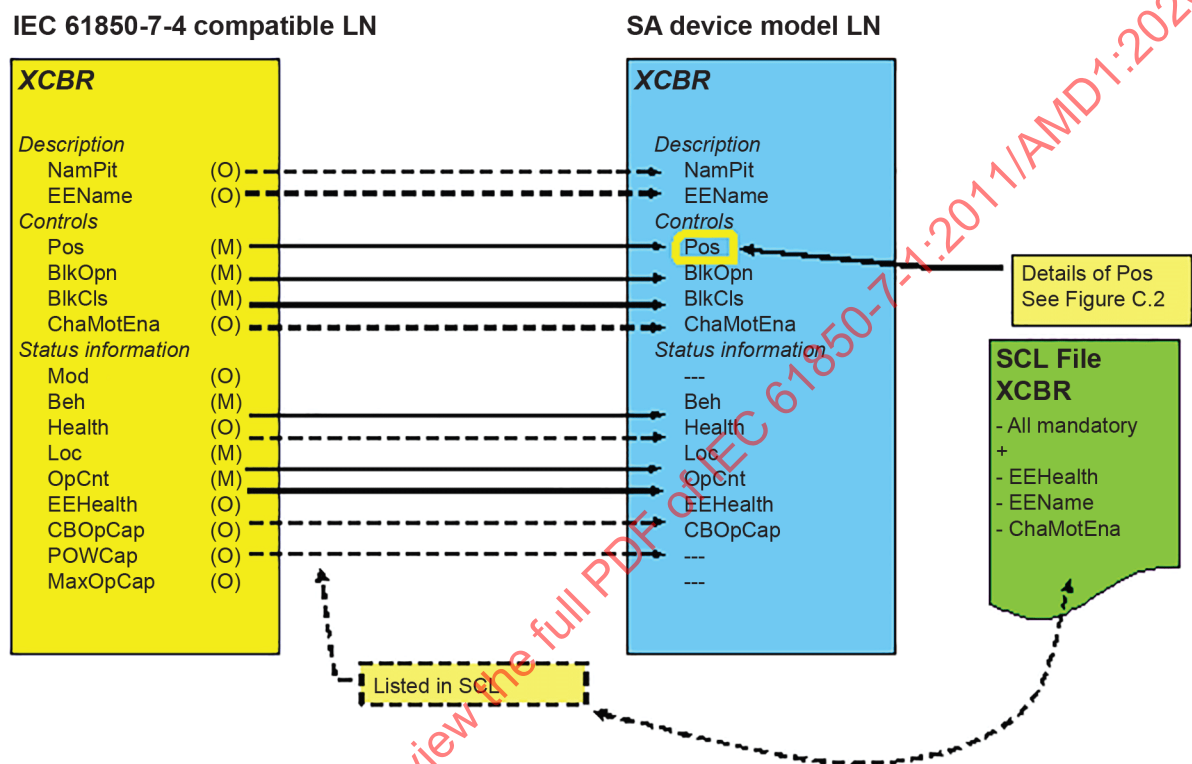
The sampled values (AmpSv and VolSv) and the corresponding ratings are used in the example. These data are referenced by the data set "DS1".

Annex C (informative)

Use of the substation configuration language (SCL)

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

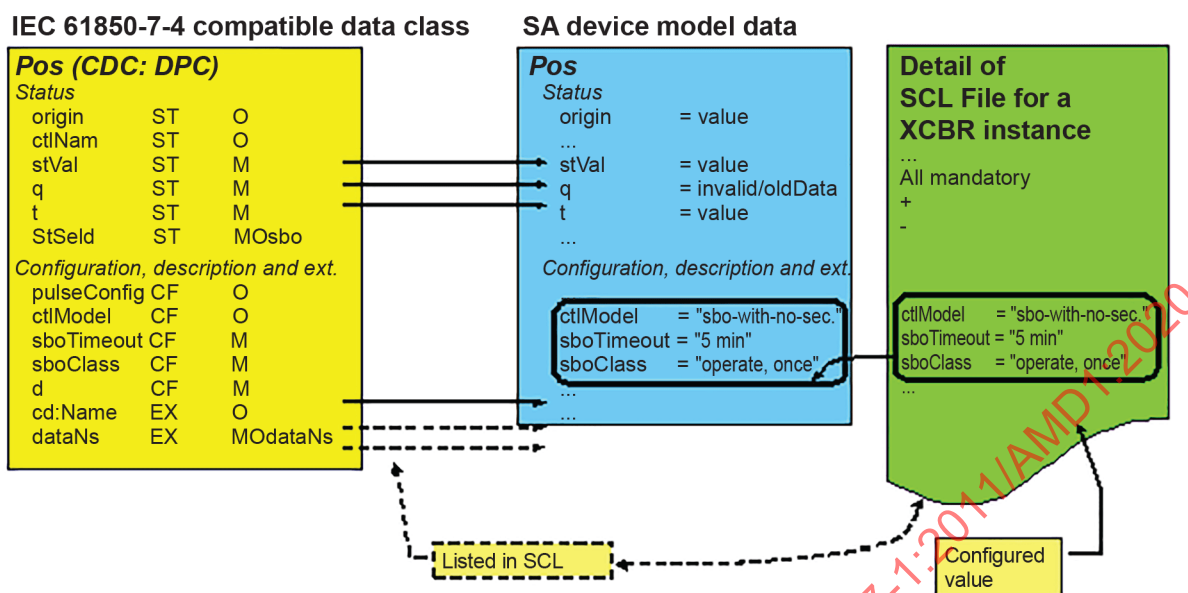
Replace existing Figure C.1 with the following new figure:



IEC

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

Replace existing Figure C.2 with the following new figure:



IEC

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

Annex D
(informative)

Applying the LN concept to options for future extensions

Replace the existing text of Annex D with the following new text:

This page is intentionally left blank.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

Annex F (informative)

Mapping the ACSI to real communication systems

Figure F.9 – MMS information report message

Replace existing Figure F.9 with the following new figure:

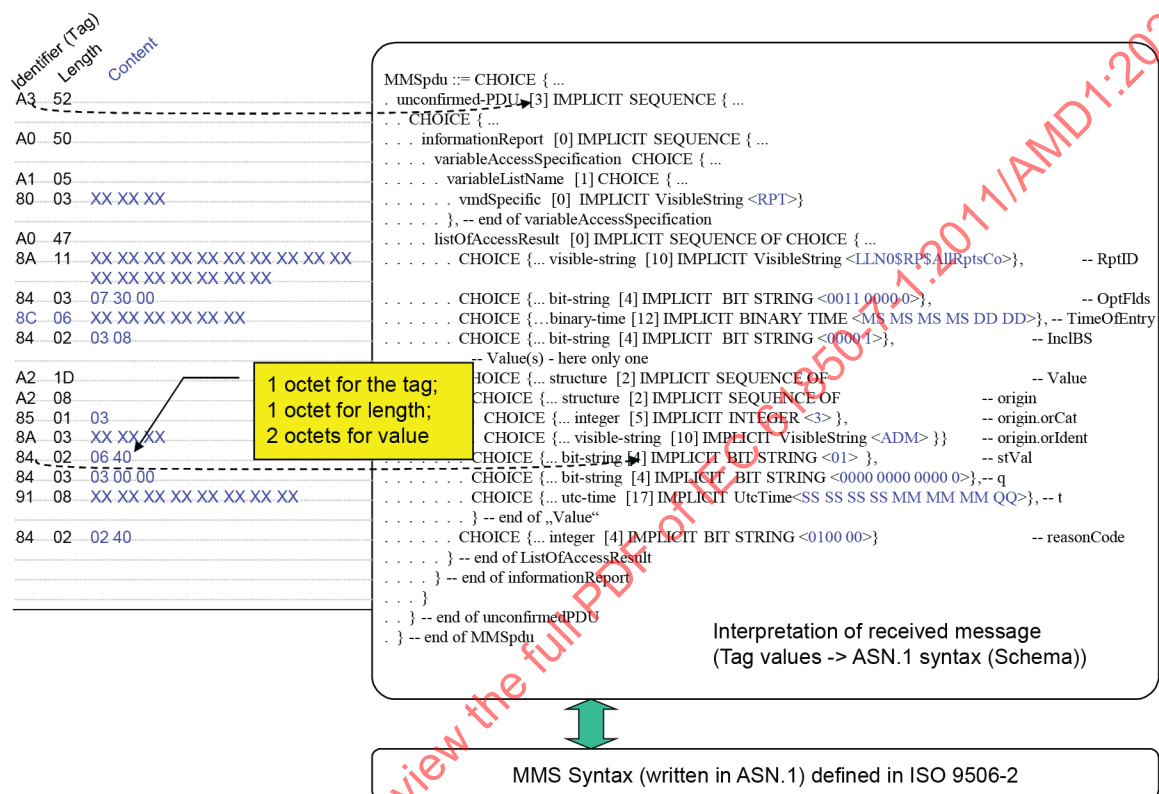


Figure F.9 – MMS information report message

Annex G (normative)

LGOS/LSVS engineering

Add, after existing Annex F, the following new Annexes G to K:

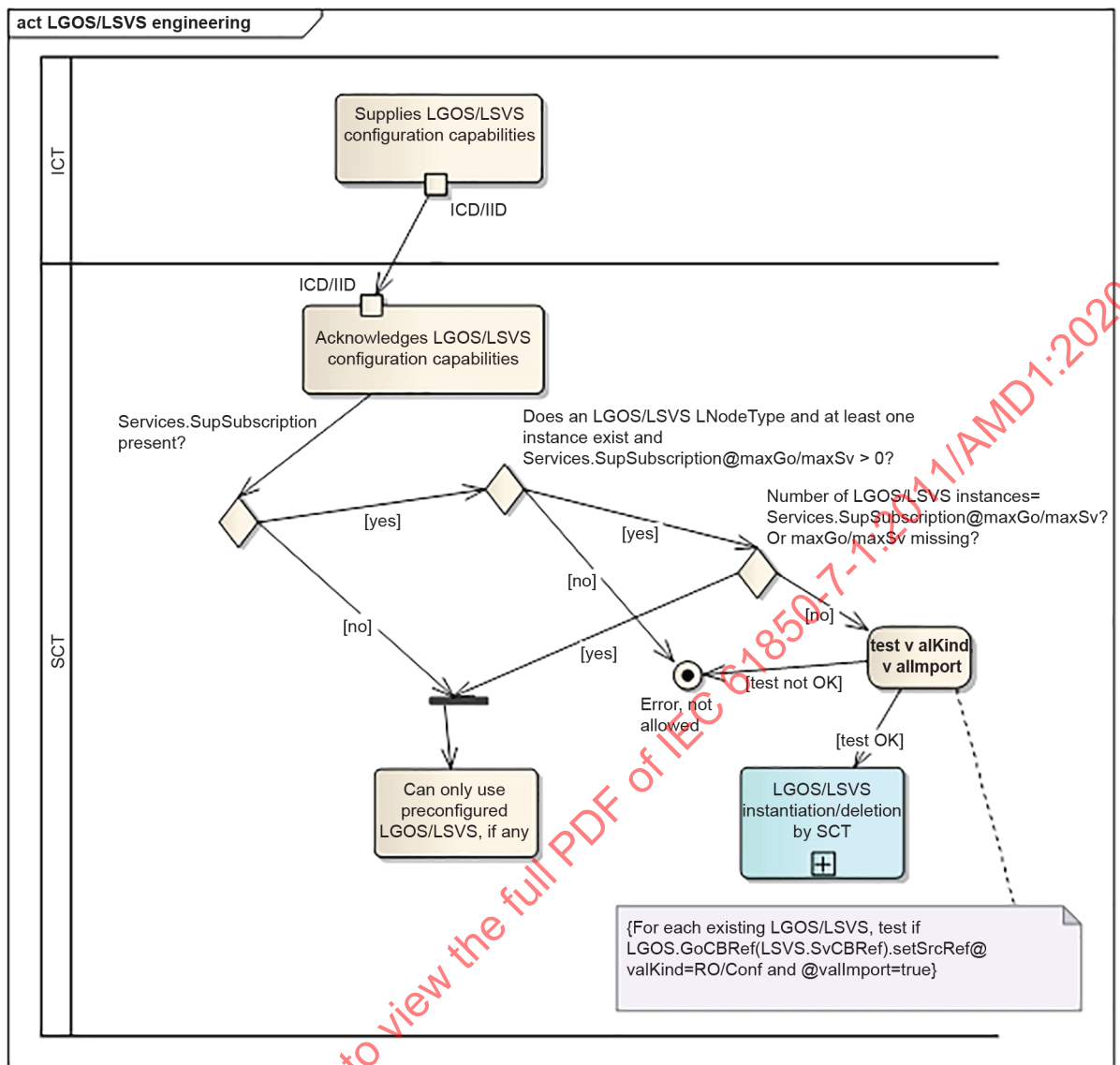
G.1 General

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.

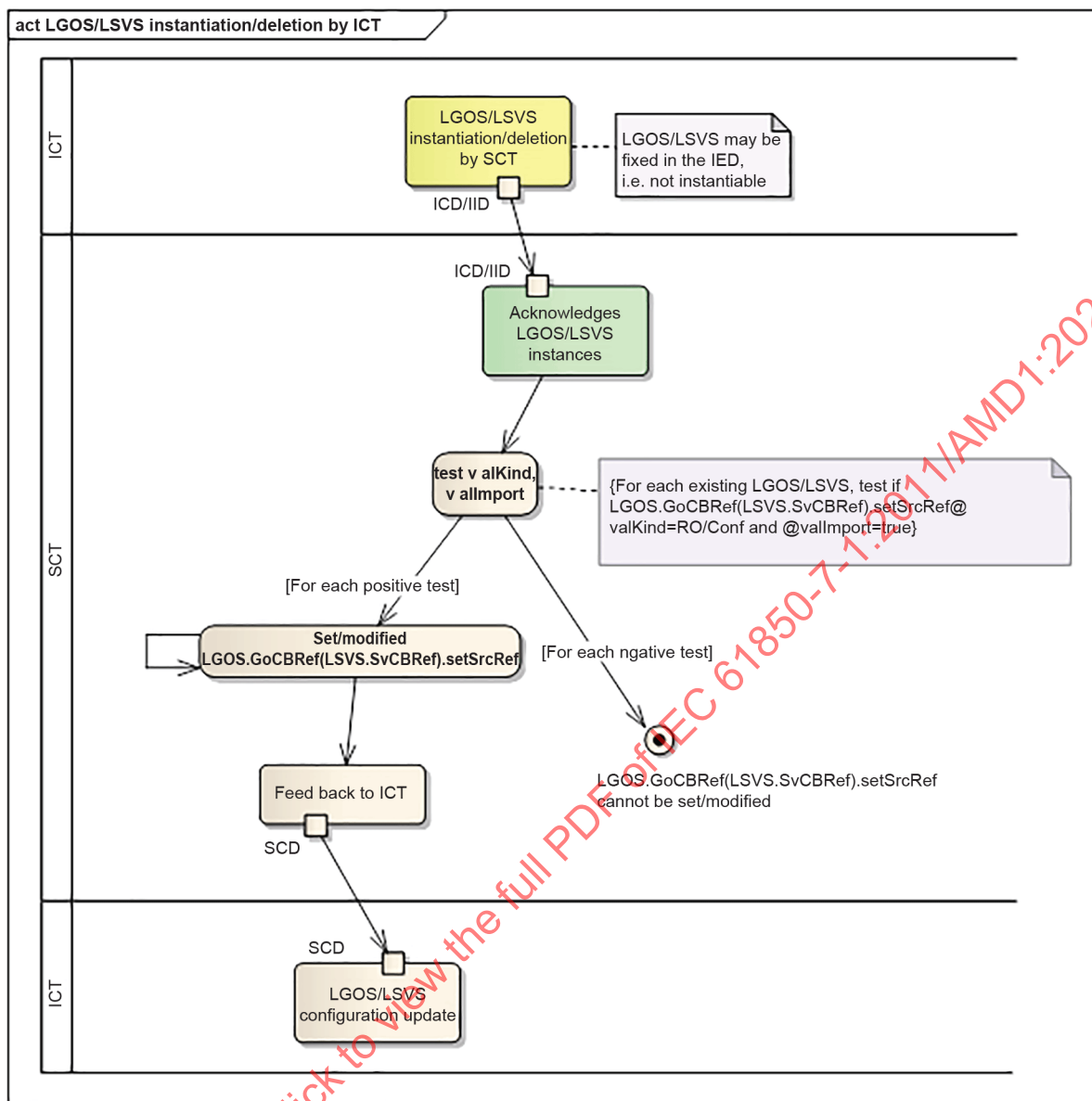


IEC

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.



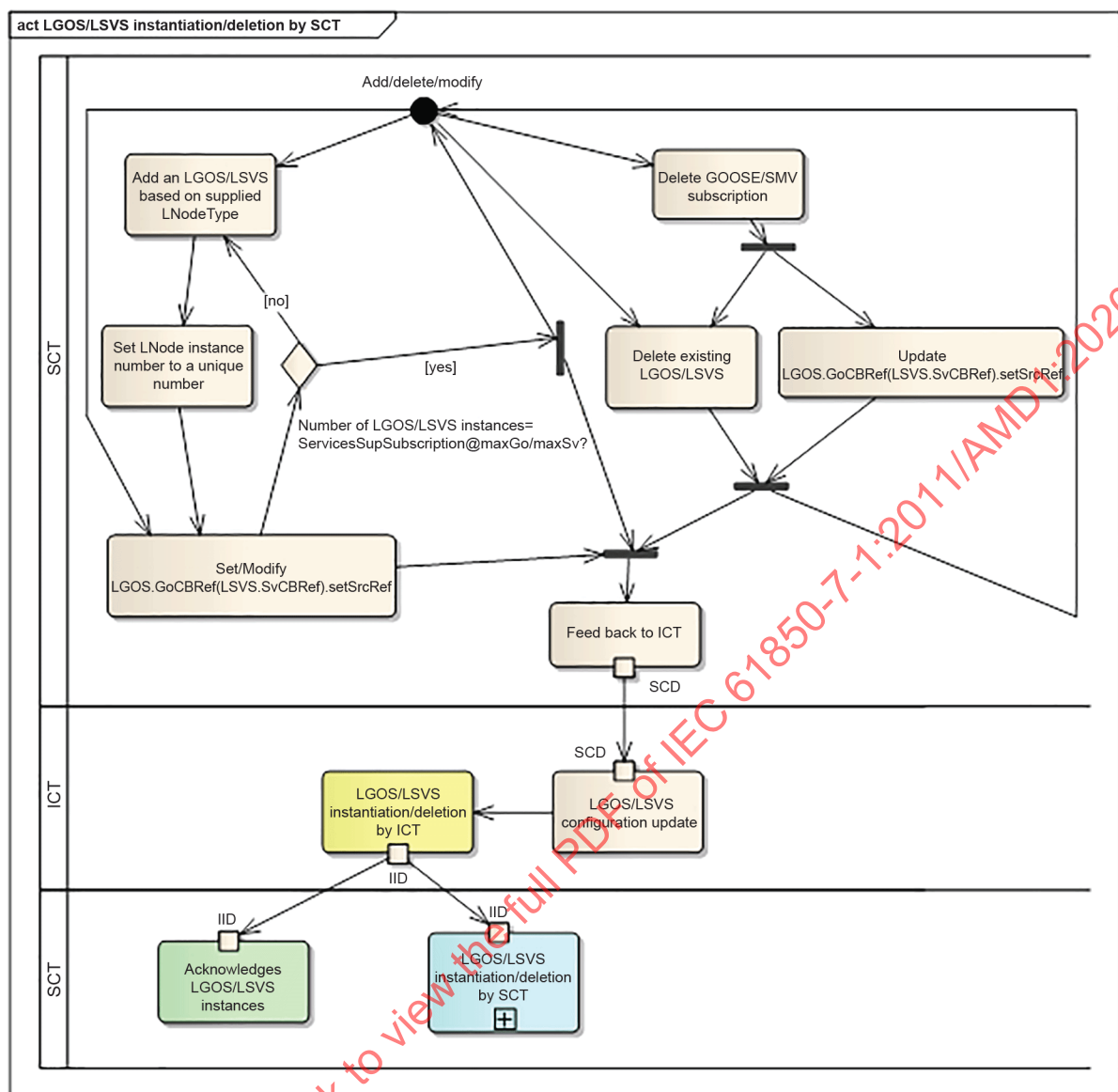
IEC

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.



IEC

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.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

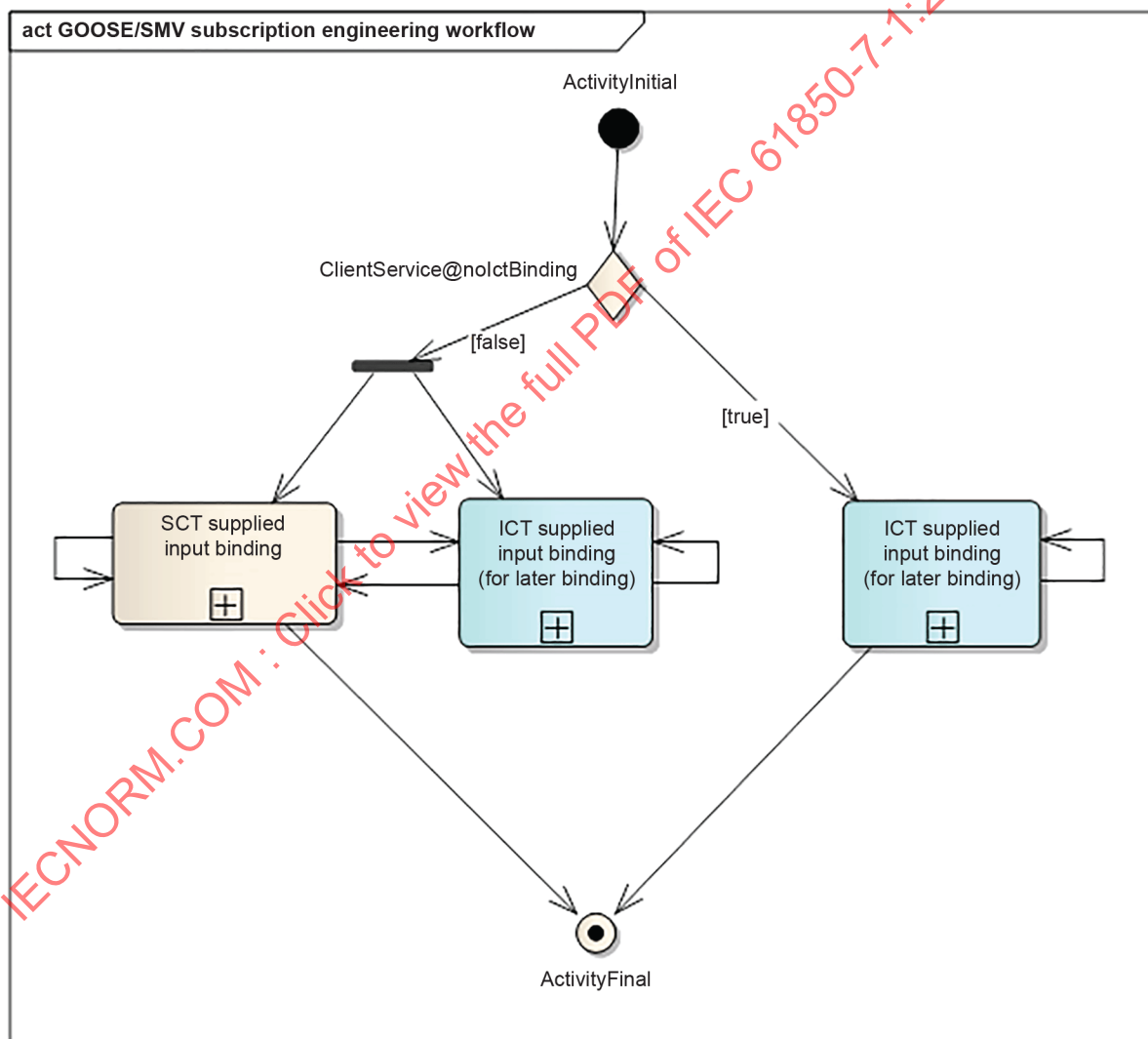
Annex H (normative)

GOOSE/SMV Subscription Configuration

H.1 General

The SCL language (IEC 61850-6) offers the capacity to document signals flow between IEDs within an PUAS. 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.



IEC

Figure H.1 – GOOSE/SMV subscription engineering workflow

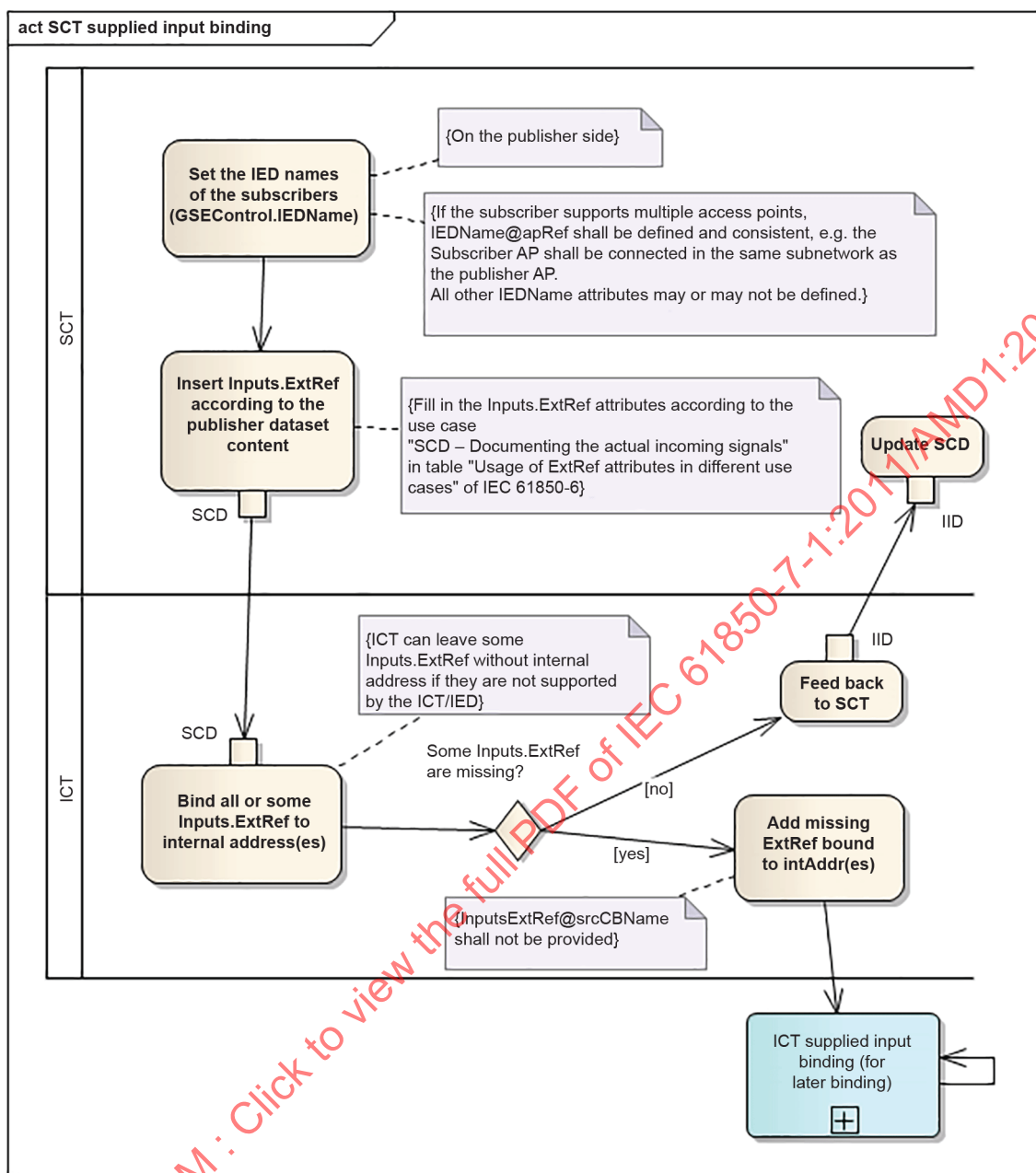
If the IED tool (ICT) has no restriction and accepts that the binding of inputs be done by the system tool (SCT), the workflow “SCT supplied input binding” applies. If not, the workflow “ICT supplied input binding” shall be followed. The capability of supporting both workflows or only the “ICT supplied 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.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

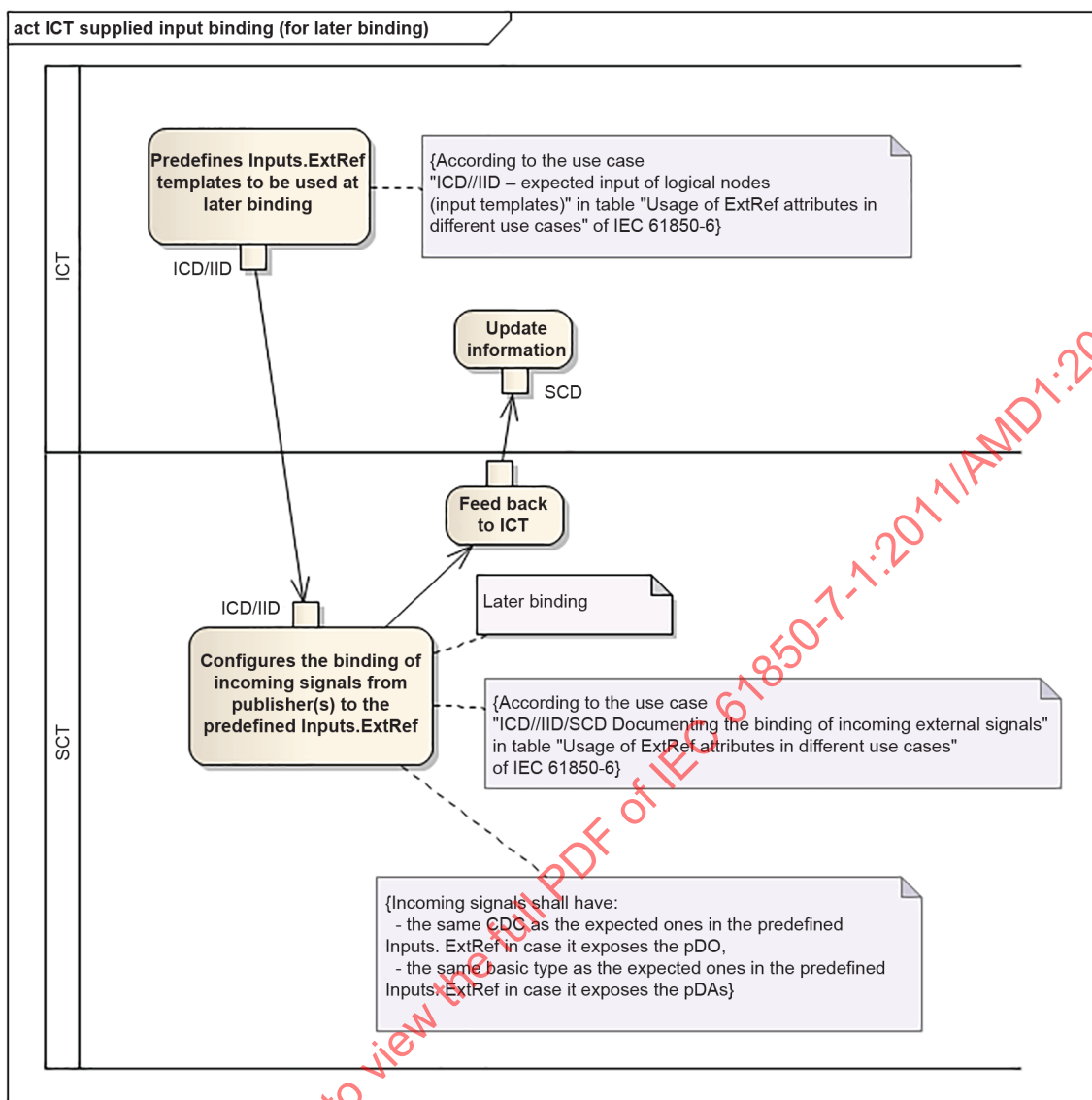


IEC

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 H.3. Note that the workflow does not show the configuration data transfer to the actual IED.



IEC

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

Annex J (normative)

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

J.1 General

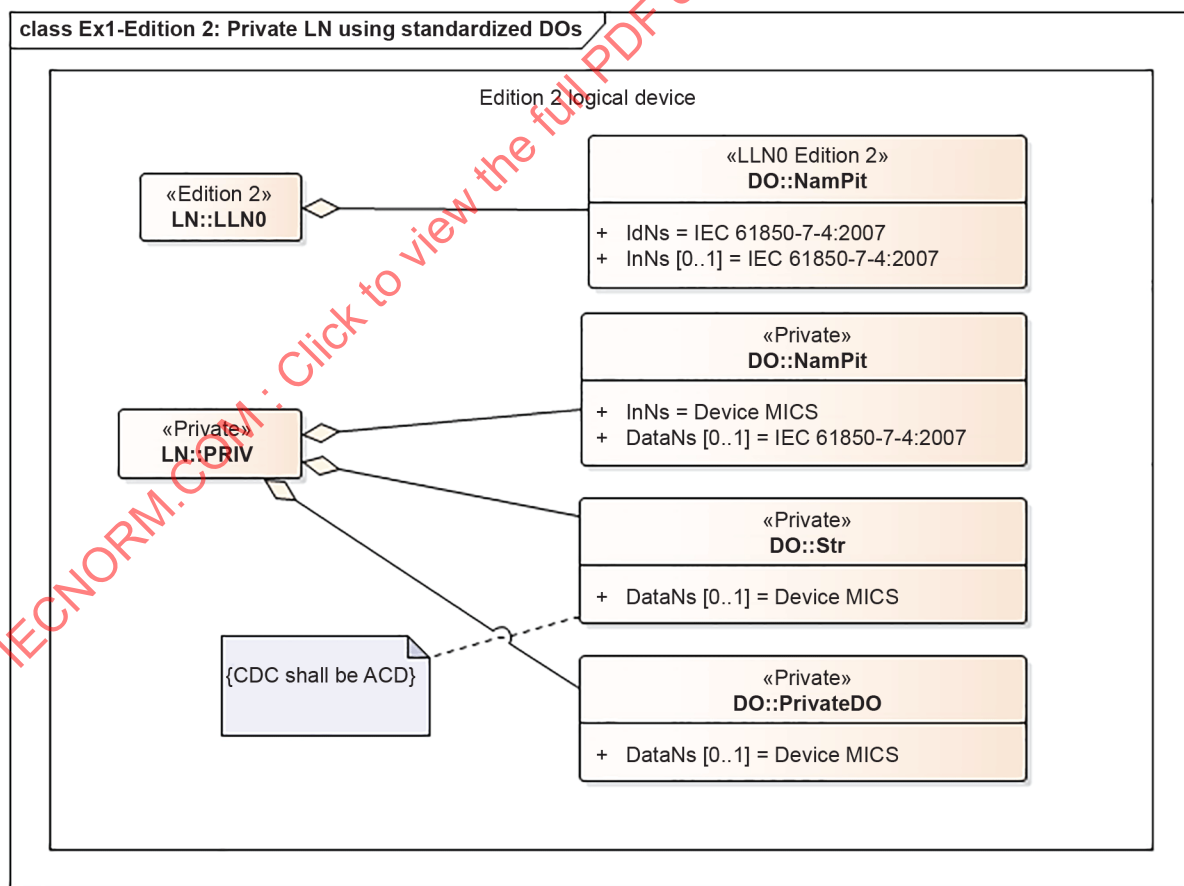
This annex contains class diagrams to illustrate the mandatory/optional use of the different namespace attributes defined in Subclause 13.5. The following definitions are used.

- 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.



IEC

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.

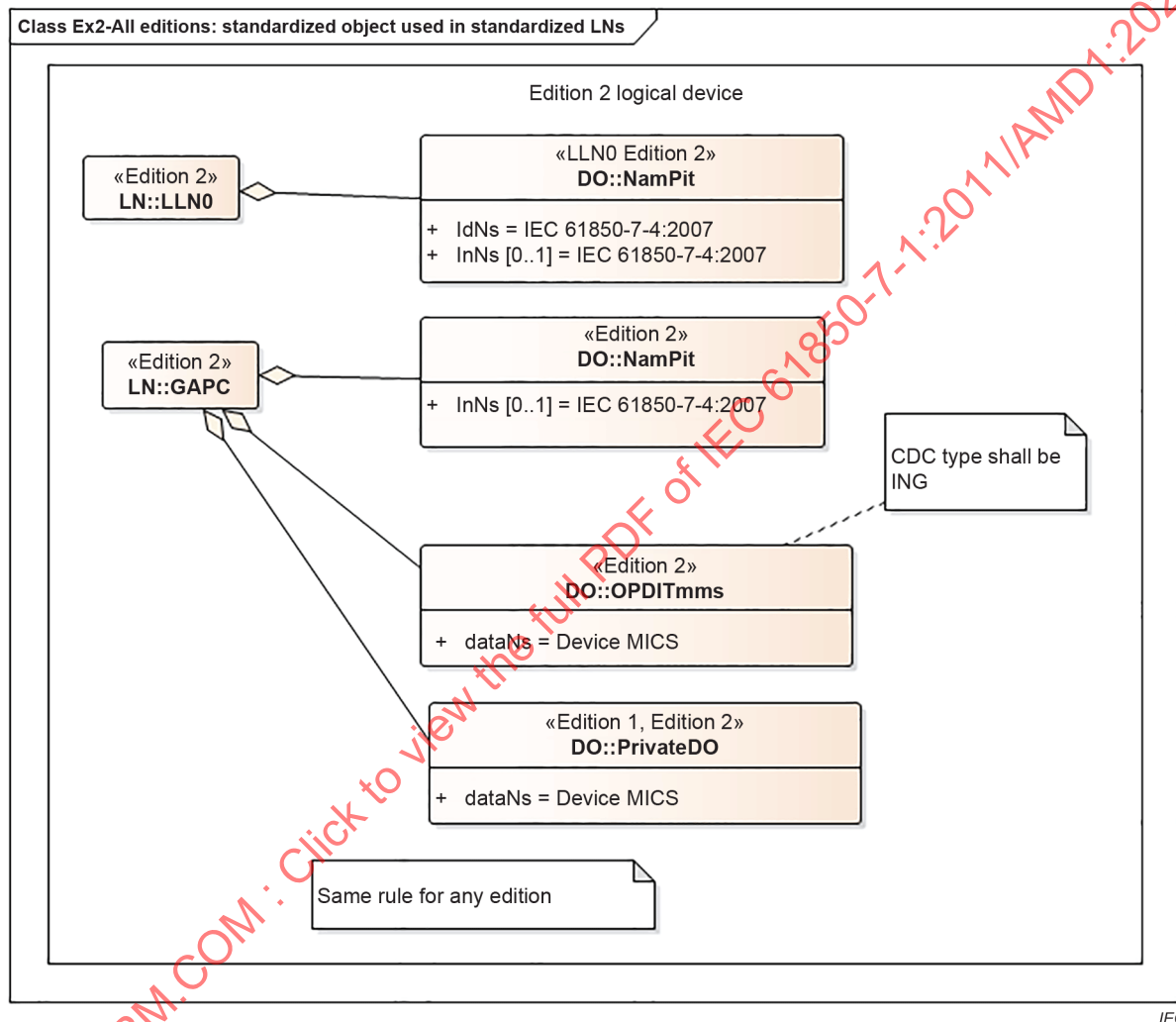
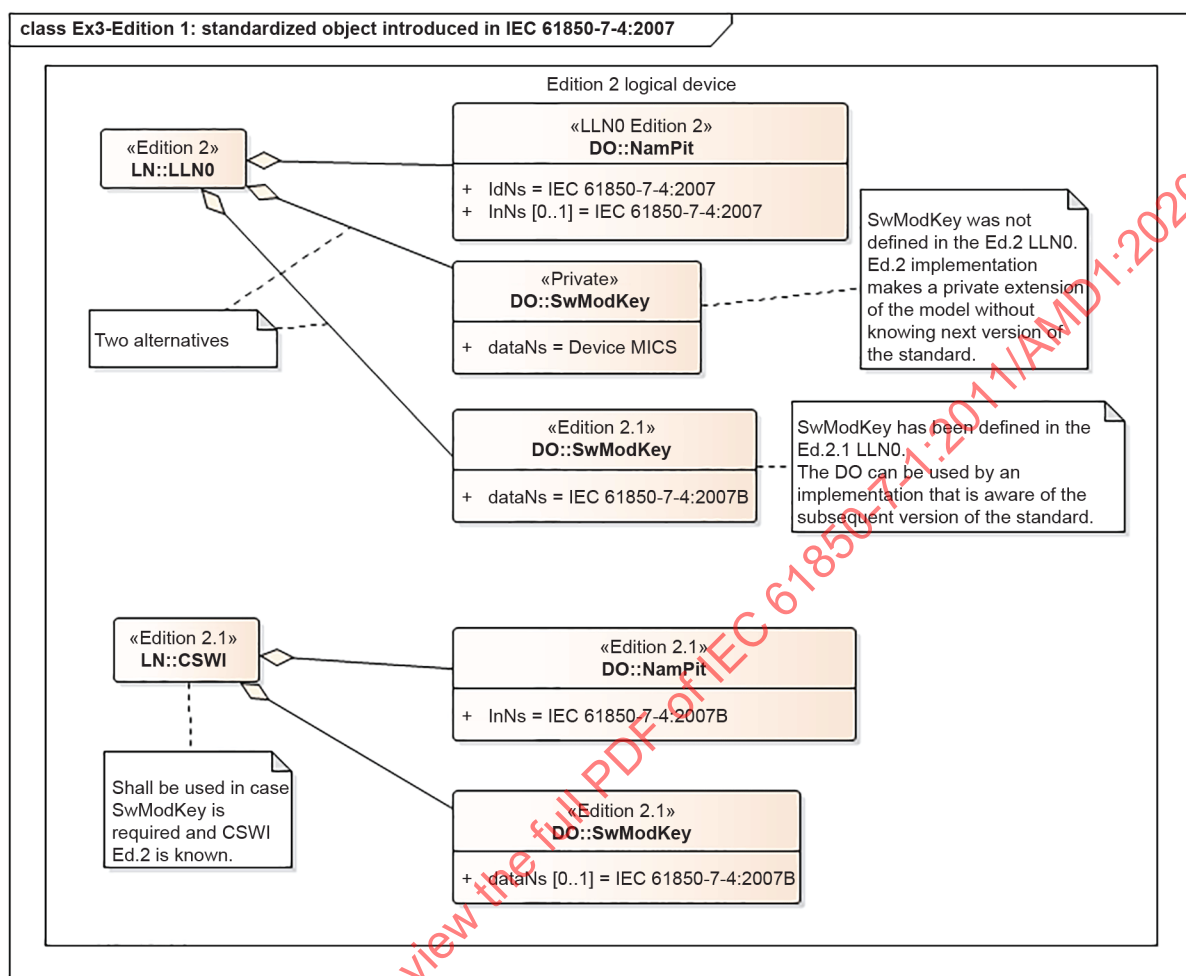


Figure J.2 –Standardized data objects used in standardized LNs

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.



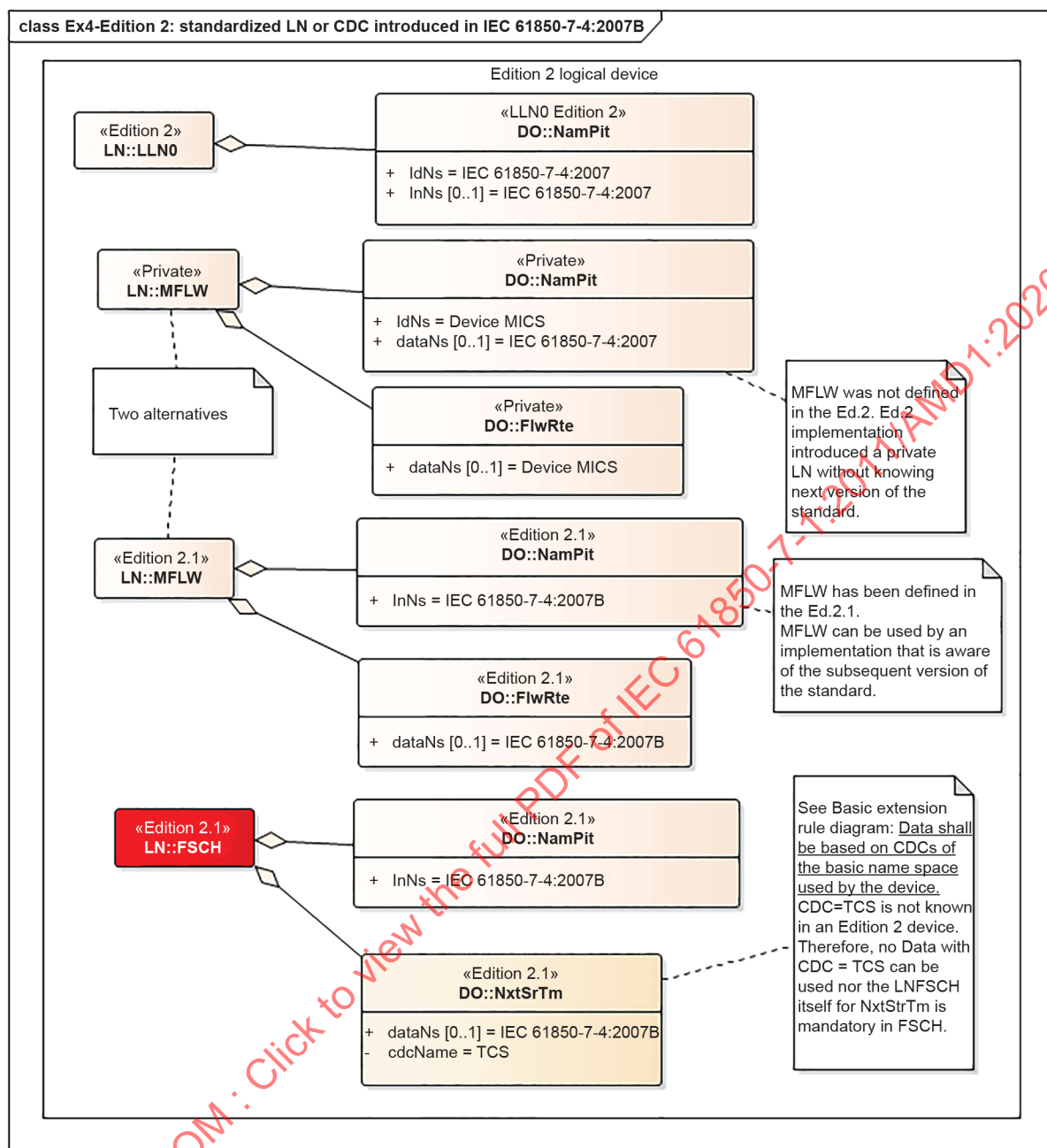
IEC

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).

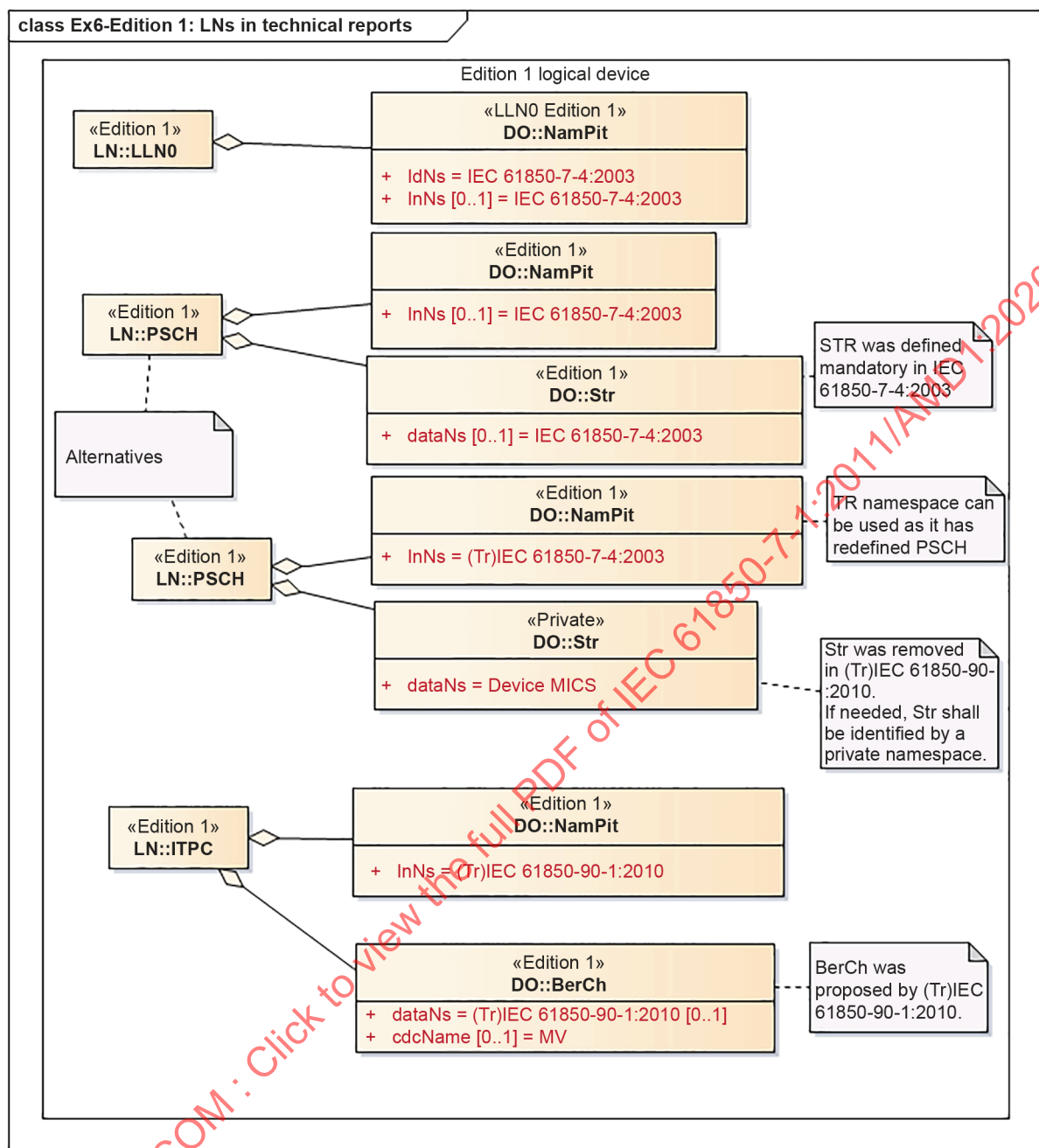


IEC

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.



IEC

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

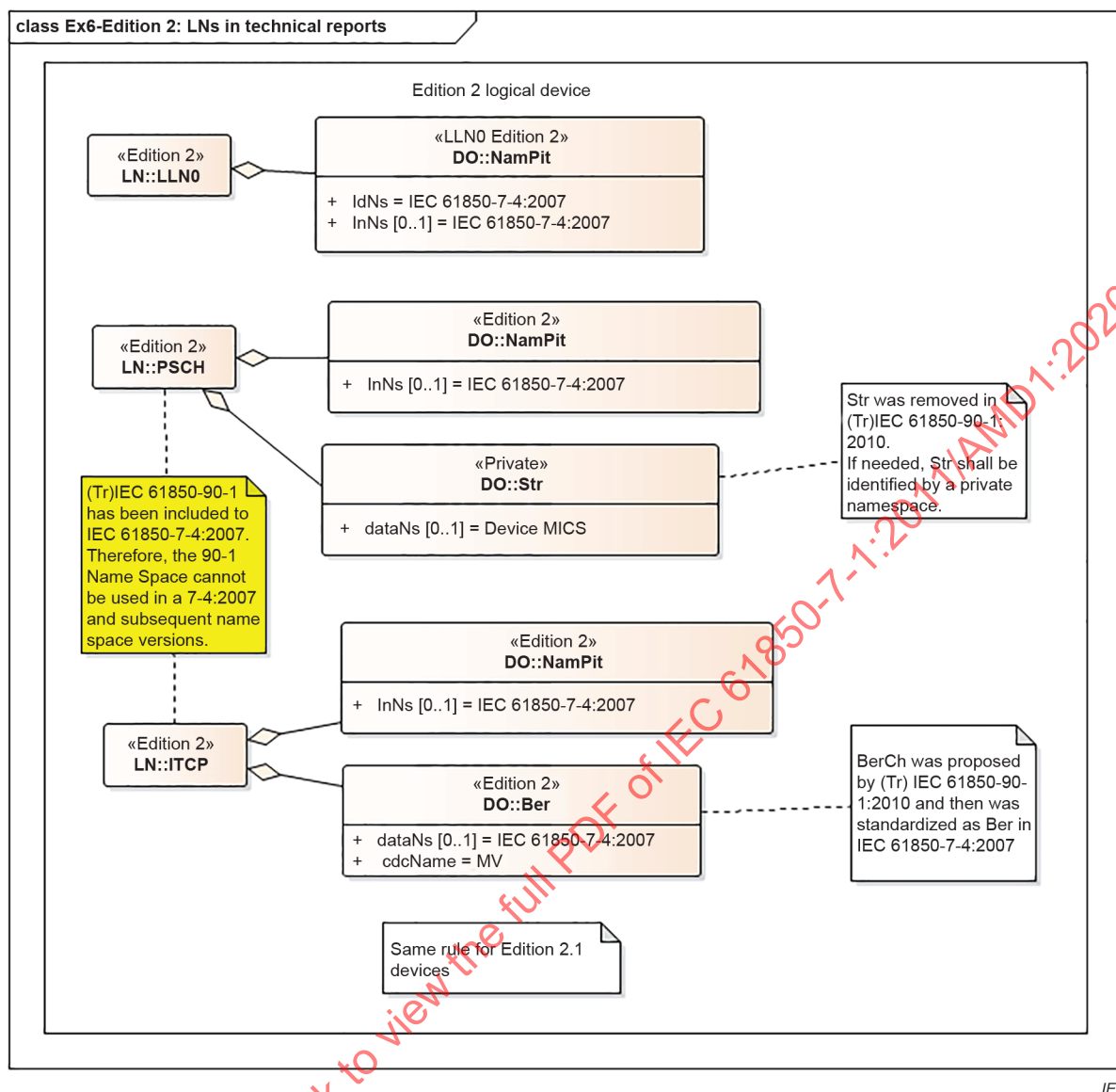


Figure J.6 – Edition 2 device: Logical nodes in technical reports

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.

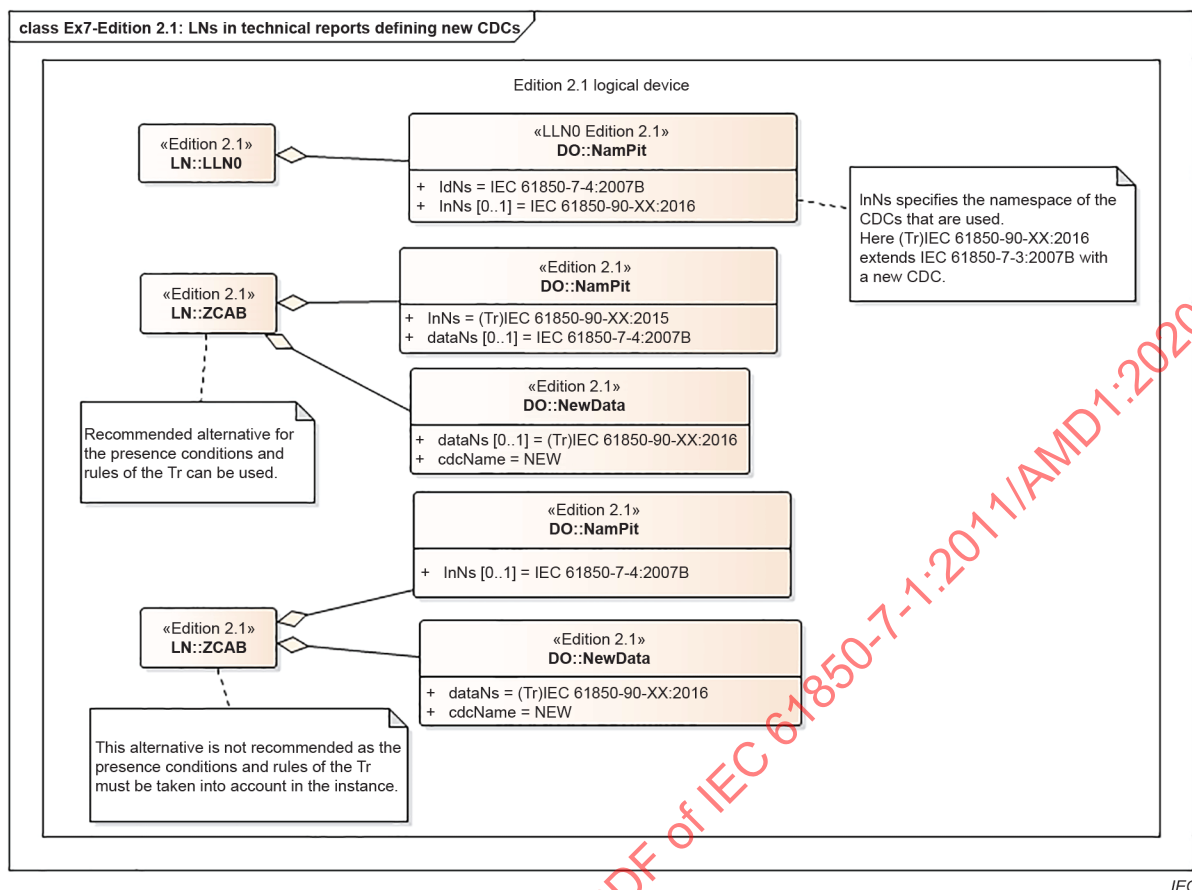


Figure J.7 – Edition 2.1 device: Logical nodes in technical reports defining new CDCs

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 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.

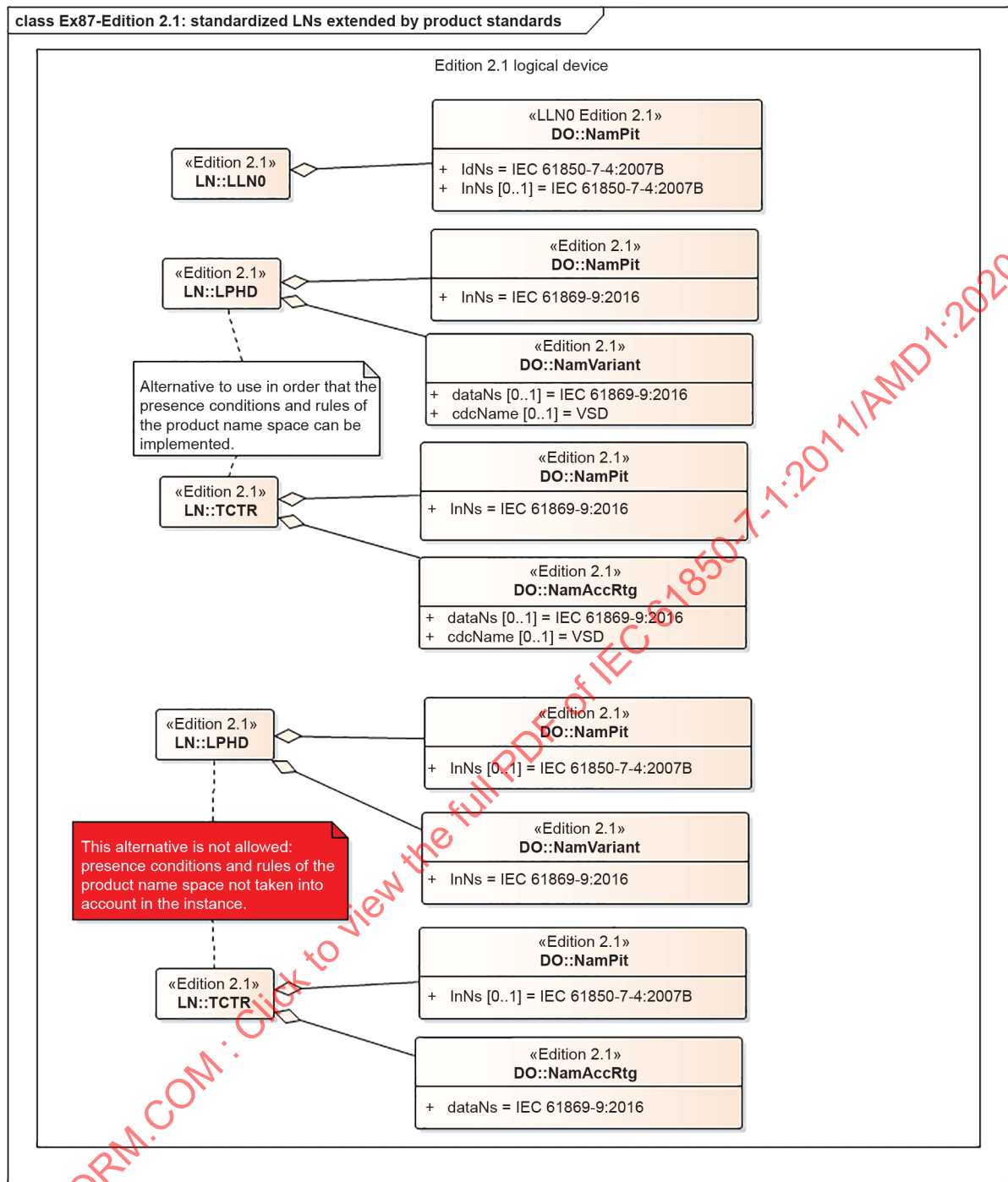
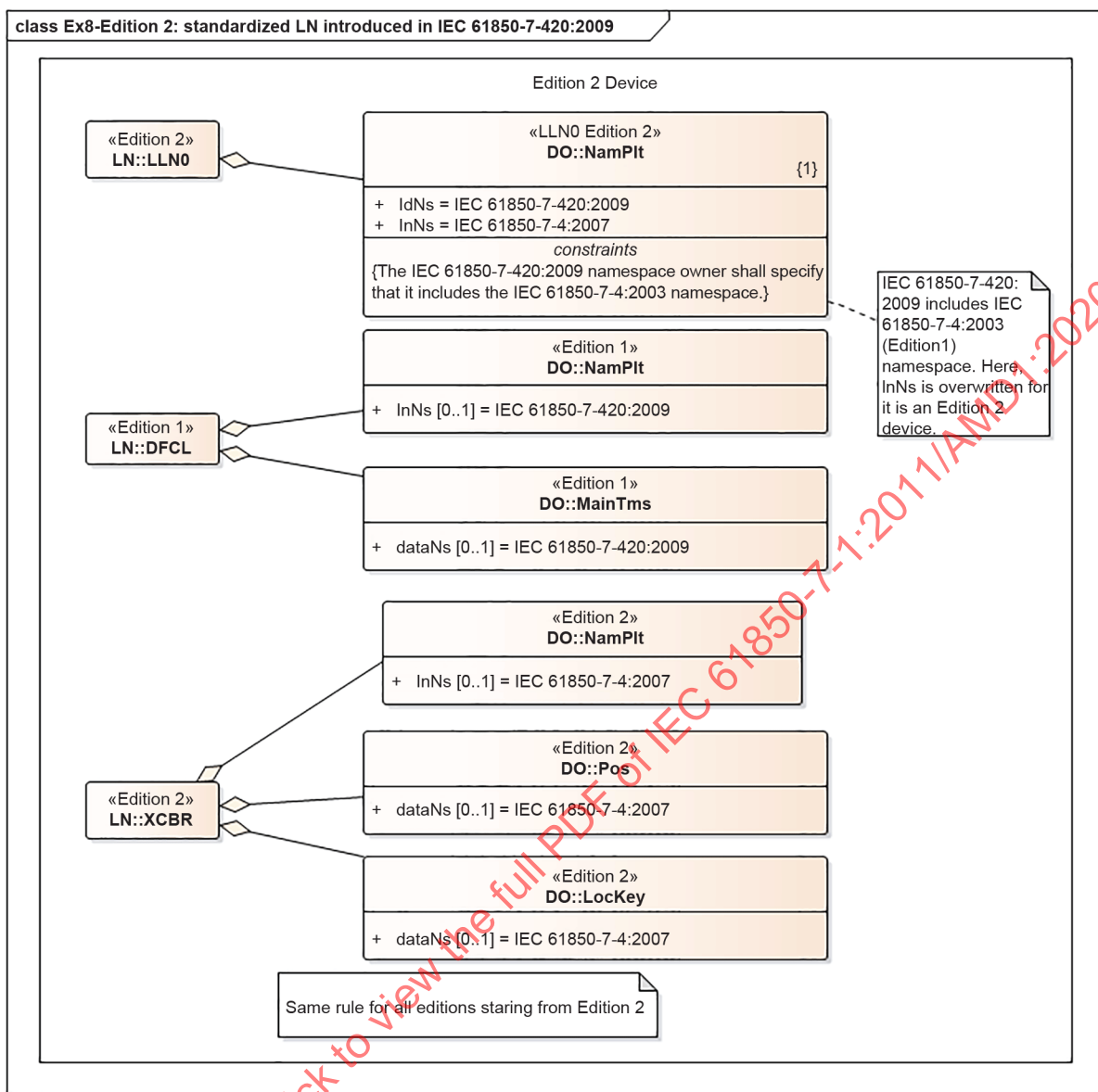


Figure J.8 – Edition 2.1 device: standardized LNs extended by other domains

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.



IEC

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

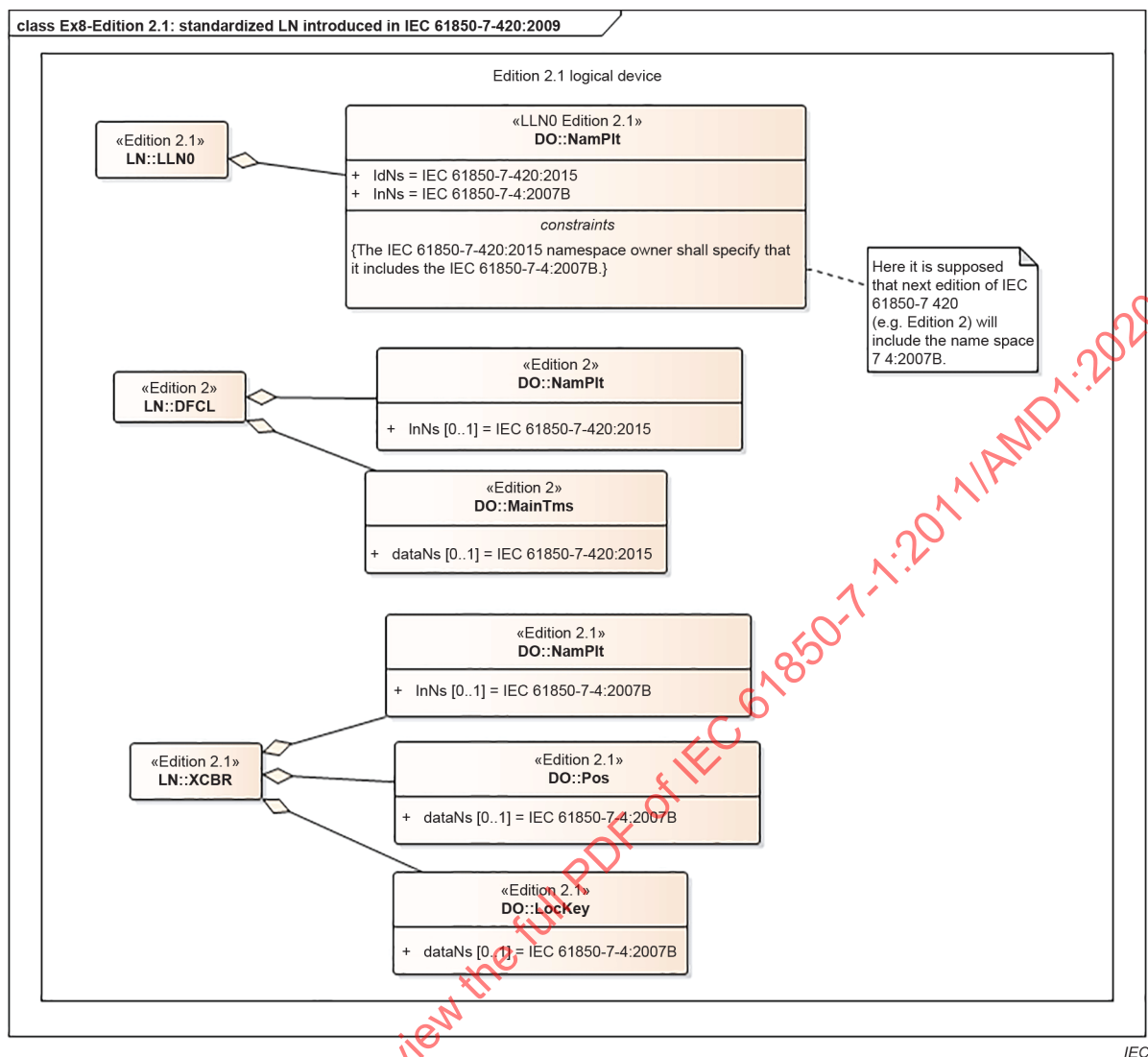


Figure J.10 – Edition 2.1 device: standardized LN introduced in IEC 61850-7-420:2015

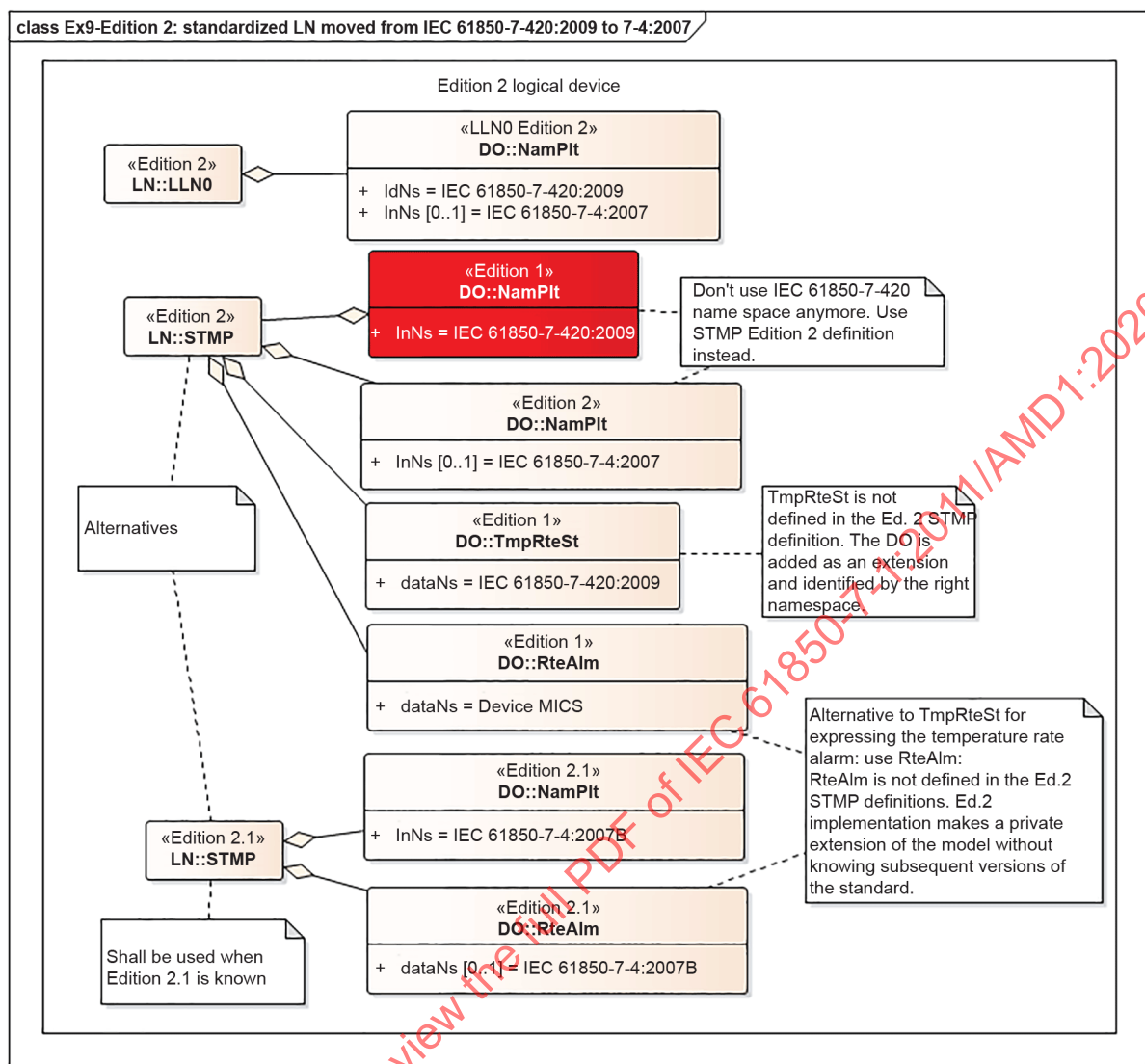
NOTE The version for the second edition of the IEC 61850-7-420 is unknown at this time, and the 2015 is used as an example.

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.



IEC

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

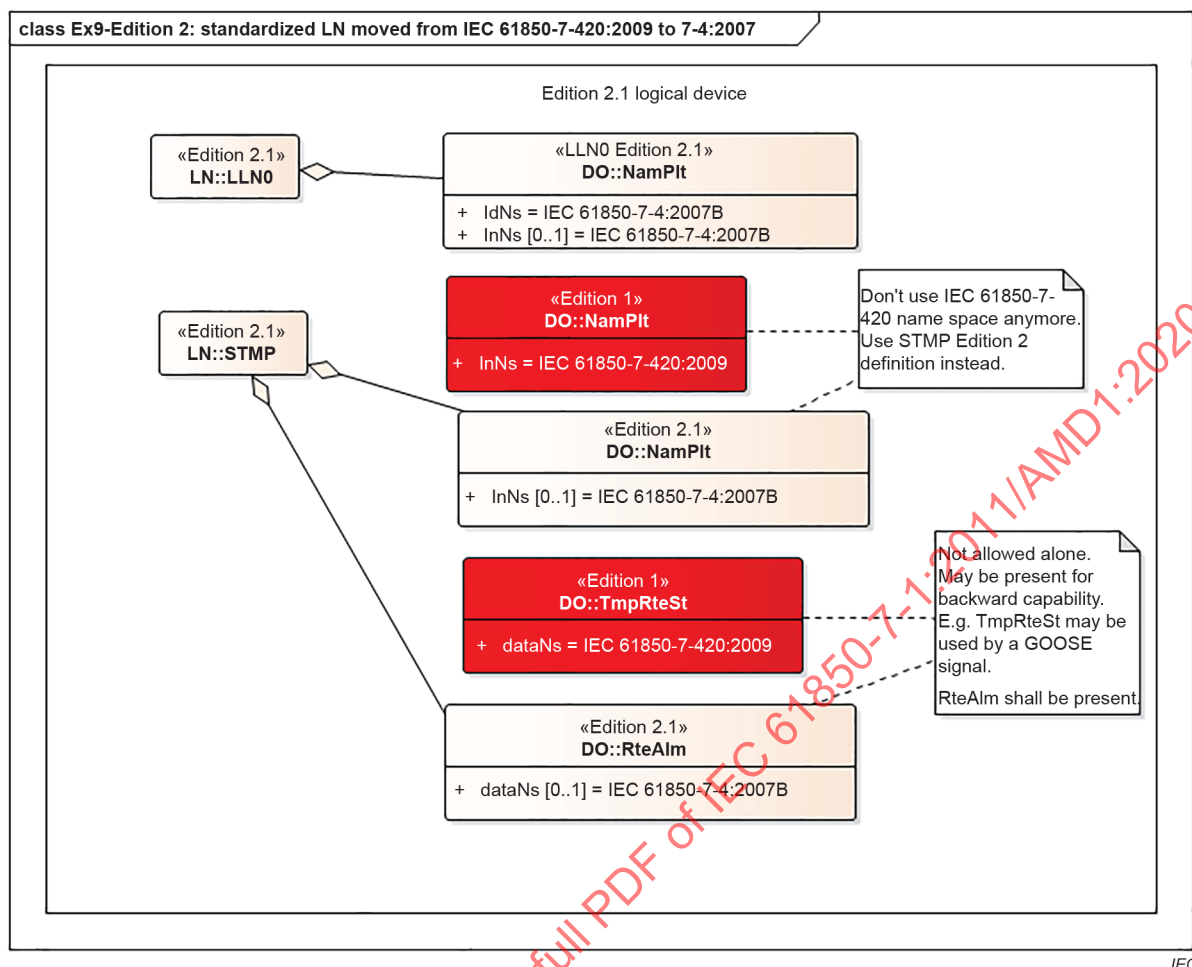


Figure J.12 – Edition 2.1 device: standardized LN moved from introduced in IEC 61850-7-420:2009 to IEC 61850-7-4:2007

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 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 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 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 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 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 a given namespace will follow a specific treatment to manage smoothly the transition:

- a deprecated element is shown as deprecated in the edition/version where the decision to deprecate is expressed,

- a deprecated element should be kept present in any associated amendment and corrigenda of the same edition,
- a deprecated element should be removed in the editions which follows (in order to not confuse the users and/or encourage some remaining usages of the deprecated element).

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 related to 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
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		Rename a DO	7-4
7		Rename a DA, Sub DO or Sub DA	
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

Use Case Number		Use Case Description	Part
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
		<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
35		<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
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 block 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	<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 Compatibility rule: The client, while browsing a data model, shall be able to ignore elements of an unknown type. <i>Backwards compatibility:</i> Guaranteed, if DA is optional; achievable if extension DA is mandatory Requires a specific compatibility rule if DA is mandatory
Subscriber	<i>Forward compatibility:</i> Not achievable. <i>Backwards compatibility:</i> Guaranteed.
IED tool of information user	<i>Forward compatibility:</i> Not achievable, if the IED associated to the tool has to process the element with the new type; otherwise guaranteed. <i>Backwards compatibility</i> Guaranteed
System tool	<i>Forward compatibility:</i> Not achievable if the system configuration requires data flow engineering involving element with the new type; otherwise guaranteed A tool shall not support data flow engineering involving elements including unknown basic types, <i>Backwards compatibility:</i> Guaranteed
System design rule	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

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 K.1.4).

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	<i>Forward compatibility: achievable</i> Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC. Compatibility rule: The client shall be able to ignore DAs with unknown FC <i>Backwards compatibility: guaranteed, if DA is optional; achievable if extension DA is mandatory</i> Requires a specific compatibility rule if DA is mandatory
Subscriber	<i>Forward compatibility: achievable</i> 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 <i>Backwards compatibility: guaranteed</i>
IED tool of information user	<i>Forward compatibility: guaranteed</i> <i>Backwards compatibility: guaranteed</i>
System tool	<i>Forward compatibility: guaranteed</i> NOTE A system tool is able to import any SCL files that contain DOs with CDCs extended with new DAs of existing basic type. <i>Backwards compatibility: guaranteed</i>
Downgrading	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.

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	<i>Forward compatibility:</i> achievable Compatibility rule: The client shall be able to use GetDataDefinition or the SCL file to learn the new structure of the CDC. <i>Backwards compatibility:</i> guaranteed, if DA is optional; achievable if DA is mandatory Requires a specific compatibility rule if DA is mandatory
Subscriber	<i>Forward compatibility:</i> achievable 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 <i>Backwards compatibility:</i> guaranteed
IED tool of information user	<i>Forward compatibility:</i> achievable Compatibility rule: the IED tool shall use ExtRef.intAddr empty to expose that a binding of an unknown DA is not supported. <i>Backwards compatibility:</i> guaranteed
System tool	<i>Forward compatibility:</i> guaranteed 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. <i>Backwards 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.
Downgrading	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.

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	<i>Forward compatibility:</i> achievable 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. <i>Backwards compatibility:</i> not relevant
Subscriber	<i>Forward compatibility:</i> achievable 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. <i>Backwards compatibility:</i> guaranteed
IED tool of information user	<i>Forward compatibility:</i> achievable. 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. <i>Backwards compatibility:</i> guaranteed
System tool	<i>Forward compatibility:</i> guaranteed <i>Backwards compatibility:</i> guaranteed
Downgrading	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.

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	<i>Forward compatibility:</i> achievable 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. <i>Backwards compatibility:</i> achievable 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. If DA is mandatory, a specific compatibility rule may be required
Subscriber	<i>Forward compatibility:</i> achievable 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. <i>Backward compatibility:</i> achievable Requires a specific compatibility rule.
IED tool of information user	<i>Forward compatibility:</i> achievable. Compatibility rule: the IED tool shall use ExtRef.intAddr empty to expose that a binding of an renamed (unknown) DA is not supported. <i>Backwards compatibility:</i> achievable Requires a specific compatibility rule
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> 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
Downgrading rule	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. NOTE: This is only useful for tools that use the SCL model; it will not help a client that browses the model

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	<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) <i>Backwards compatibility:</i> achievable 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
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 if DAs with deprecated FC were only optional, achievable if at least one DA was mandatory 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

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	<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) <i>Backwards compatibility:</i> guaranteed, if the client still supports the deprecated DA, otherwise achievable 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.
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 if DA was optional, achievable if DA was mandatory 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.2.10 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	<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) <i>Backwards compatibility:</i> achievable 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
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 if DA was optional, achievable if DA was mandatory 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.2.11 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	<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 with generic rule that client shall be tolerant for missing elements (see clause K.1.5) <i>Backward compatibility:</i> guaranteed

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 known by the old server will receive a negative response.

Enumerations 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 old 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	<i>Forward compatibility:</i> achievable Compatibility rule: The client shall ignore the new control blocks classes that it may have discovered using the GetLogicalNodeDirectory. <i>Backwards compatibility:</i> guaranteed.
Subscriber	<i>Forward compatibility:</i> not relevant <i>Backwards compatibility:</i> not relevant

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	<i>Forward compatibility</i> : not relevant <i>Backwards compatibility</i> : guaranteed
Subscriber	<i>Forward compatibility</i> : not achievable <i>Backwards compatibility</i> : guaranteed. New services are never mandatory, and their support shall be 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.
IED tool of service caller	<i>Forward compatibility</i> : not achievable <i>Backwards compatibility</i> : guaranteed.
System tool	<i>Forward compatibility</i> : Not relevant. <i>Backwards compatibility</i> : Not relevant.
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 SGC.B.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	<i>Forward compatibility</i> : achievable Compatibility rule: The client ignores the new control block attributes it discovers using the Get<ControlBlock>Value(). <i>Backwards compatibility</i> : guaranteed, if control block attribute is optional; achievable if control block attribute is mandatory Requires a specific compatibility rule if control block attribute is mandatory.
Subscriber	<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. <i>Backwards compatibility</i> : guaranteed, if control block attribute is optional; achievable if control block attribute is mandatory Requires a specific compatibility rule if control block attribute is mandatory.

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	<i>Forward compatibility</i> : guaranteed. <i>Backwards compatibility</i> : achievable. See K.1.5.3.
Subscriber	<i>Forward compatibility</i> : not achievable if the deprecated control block class is associated to the subscribed publisher process. Otherwise guaranteed. <i>Backwards compatibility</i> : achievable. See K.1.5.3.

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	<i>Forward compatibility</i> : achievable. CRu: client shall implement functionality associated to all control block attributes. <i>Backwards compatibility</i> : achievable Requires a specific compatibility rule if control block attribute is mandatory.
Subscriber	<i>Forward compatibility</i> : achievable. CRu: subscribers shall implement functionality associated to all control block attributes. <i>Backwards compatibility</i> : achievable. Requires a specific compatibility rule if control block attribute is mandatory.

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

Remove the following references from the bibliography:

IEC 61400 (all parts)

IEC TR 61850-90-1

IEEE-SA TR 1550

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

AVANT-PROPOS

Le présent amendement a été établi par le comité d'études 57 de l'IEC: Gestion des systèmes de puissance et échanges d'informations associés.

Le texte de cet amendement est issu des documents suivants:

FDIS	Rapport de vote
57/2201/FDIS	57/2221/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cet amendement.

Le comité a décidé que le contenu de cet amendement et de la publication de base ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

4 Abréviations

Ajouter les nouveaux termes suivants à la liste des abréviations:

MICS	Model implementation conformance statement (Déclaration de conformité de mise en œuvre du modèle)
PICS	Protocol implementation conformance statement (Déclaration de conformité de mise en œuvre du protocole)
SICS	SCL implementation conformance statement (Déclaration de conformité de mise en œuvre SCL)

5 Vue d'ensemble des concepts de la série IEC 61850

5.3 Modèles d'information des systèmes d'automatisme de poste

Ajouter la nouvelle note de bas de page à la fin du troisième alinéa sous la Figure 3:

Il existe une exception: l'application proxy/passerelle. Pour plus d'informations, voir 8.2.3 Passerelles et proxys.

5.4 Applications modélisées par des nœuds logiques définis dans l'IEC 61850-7-4

Figure 4 – Catégories d'informations de nœuds logiques

Remplacer la Figure 4 existante par la nouvelle figure suivante:

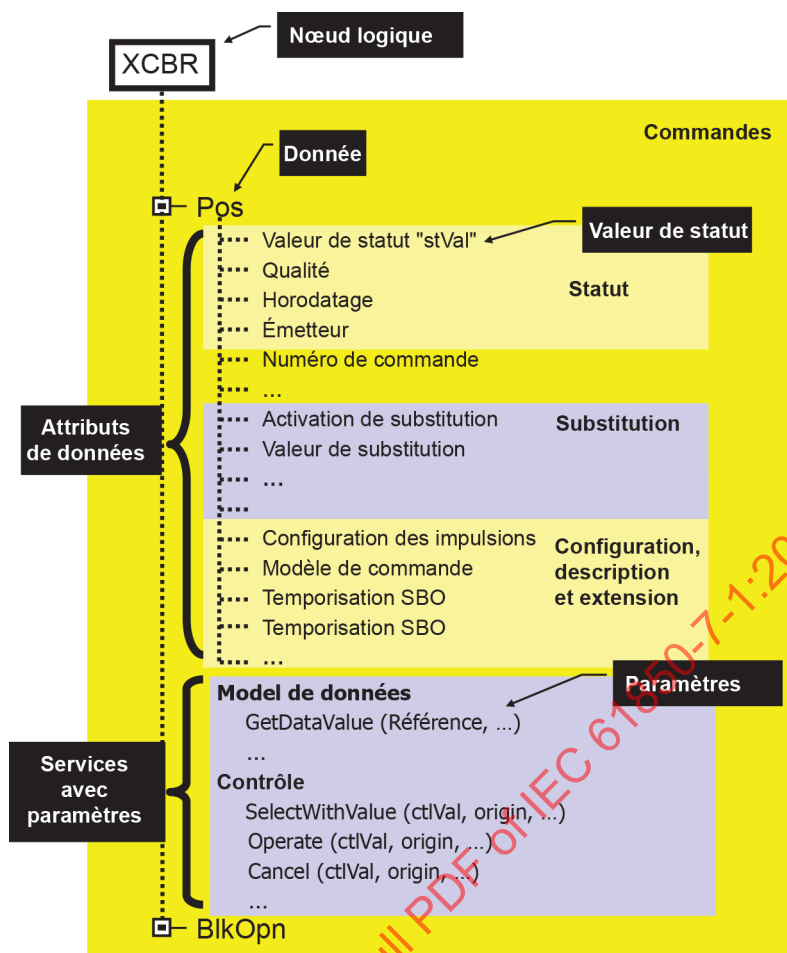


IEC

5.5 La sémantique est attachée aux données

Figure 6 – Informations de position représentées sous la forme d'un arbre (à titre conceptuel)

Remplacer la Figure 6 existante par la nouvelle figure suivante:



IEC

Remplacer le premier tableau du 5.5 par le nouveau tableau suivant:

Nom d'attribut	Type d'attribut	FC	TrgOp	(Valeur/plage de valeurs) Explication	PresCond
stVal	DpStatusKind	ST	dchg	Valeur de statut de l'objet de données contrôlable.	M
ctlModel	CtlModelKind	CF	dchg	Modèle de commande défini de l'IEC 61850-7-2 qui reflète le comportement de l'objet de données contrôlable..	M

Sous ce tableau, remplacer le dernier point de la liste par le nouveau point suivant:

- Les conditions de présence, par exemple obligatoire (M) ou facultatif (O). Ces conditions spécifient la présence d'éléments dans un contexte donné (un LN, une CDC, un type d'attribut de données ou un objet de donnée pour un dataNs) et sont spécifiées dans l'IEC 61850-7-2.

Après la liste, remplacer le premier alinéa et le tableau suivant par le nouvel alinéa suivant:

Les noms des attributs de données sont des noms normalisés (c'est-à-dire qu'ils sont réservés) qui ont une sémantique spécifique dans le contexte de la série IEC 61850. La sémantique de tous les noms d'attributs de données est définie dans l'IEC 61850-7-3.

6 Approche de modélisation de la série IEC 61850

6.2 Création de modèles d'information par composition pas à pas

Tableau 2 – Classe de nœuds logiques XCBR (à titre conceptuel).

Remplacer le Tableau 2 existant par le nouveau tableau suivant:

Descriptions
Plaque signalétique de l'équipement extérieur
Plaque signalétique du nœud logique
Commandes
Position de commutateur (voir ci-dessous pour les détails)
Blocage de l'ouverture
Blocage de la fermeture
Moteur de chargeur activé
Autorité de contrôle au niveau poste. Commute entre le niveau poste et un niveau supérieur.
...
Informations de statut
Mode
Behaviour
Health
Santé de l'équipement extérieur
Compteur de manœuvres
Fonctionnement local (indique le basculement entre le fonctionnement local et le fonctionnement distant (remote): local=TRUE, remote= FALSE)
Capacité de fonctionnement du disjoncteur
Capacité de coupure en un point de l'onde
Capacité de fonctionnement du disjoncteur à pleine charge
...

Remplacer le tableau existant sous la Figure 11 par le nouveau tableau suivant:

Nom d'objet de données	Explication
...	...
Pos	(commandable) Position des disjoncteurs/commutateurs.
...	...

6.4.2 Modèle de sortie

6.4.2.1 Concept de modèle de commande

Figure 14 – Modèle de sortie (étape 1) (à titre conceptuel)

Remplacer la Figure 14 existante par la nouvelle figure suivante:

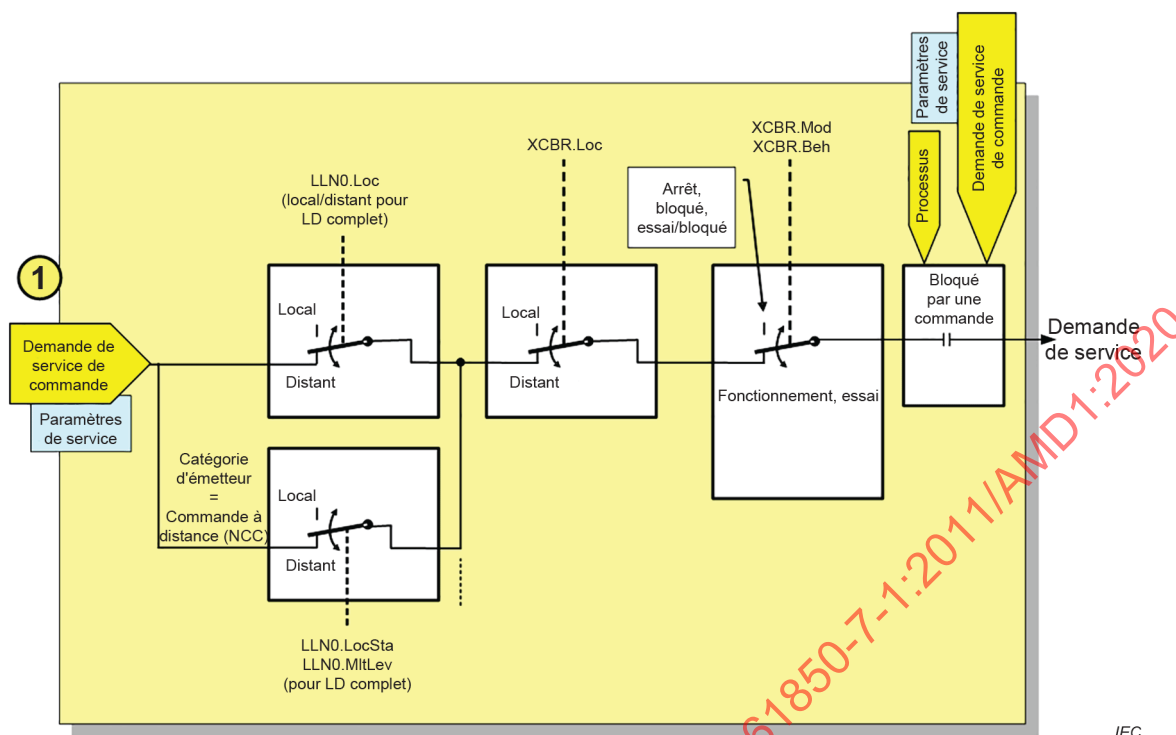


Figure 14 – Modèle de sortie (étape 1) (à titre conceptuel)

6.4.2.2 Concept de modèle GSE et SMV

Ajouter le nouveau texte suivant après l'alinéa sous la Figure 16:

L'action de recevoir des messages GOOSE et SMV par un IED est appelée "abonnement". La méthode de définition des abonnements au moyen d'outils de configuration est décrite dans l'Annexe H normative.

6.4.2.4 Données de réglage et bloc de contrôle de groupes de réglage

Tableau 3 – Extrait de réglage du statut entier (c'est-à-dire entier)

Remplacer le Tableau 3 existant par le nouveau tableau suivant:

Classe ING					
Nom d'attribut	Type d'attribut	FC	TrgOp	Valeur/plage de valeurs	PresCond
DataAttribute pour configuration, description et extension					
réglage					
setVal	INT32	SP	dchg	Valeur du réglage de statut.	AtMostOne
setVal	INT32	SG, SE		Valeur du réglage de statut.	AtMostOne
configuration, description et extension					
minVal	INT32	CF	dchg	Réglage minimal pour 'setVal'.	O
maxVal	INT32	CF	dchg	Réglage maximal pour 'setVal'.	O
stepSize	INT32U	CF	dchg	(range=[1...(maxVal-minVal)]) Etape entre les valeurs individuelles de 'setVal'.	O
units	Unité	CF	dchg	Unité pour 'setVal', 'minVal', 'maxVal', 'stepSize'.	O
d	VISIBLE STRING255	CC		Texte descriptif des données. Si elle est utilisée dans la CDC LPL, la description fait référence au nœud logique.	O
...	...	...			...

6.4.3 Modèle d'entrée

6.4.3.2 Traitement et surveillance des valeurs d'attribut de données et détection d'événement

Remplacer la dernière phrase du deuxième alinéa par la nouvelle phrase suivante:

La valeur de mag doit être mise à jour à la valeur actuelle de instMag lorsque la valeur a été modifiée en fonction de la valeur des paramètres de configuration db et dbRef de la donnée concernée.

Remplacer la Figure 19 existante et l'alinéa qui suit par la nouvelle figure et le nouveau texte suivants:

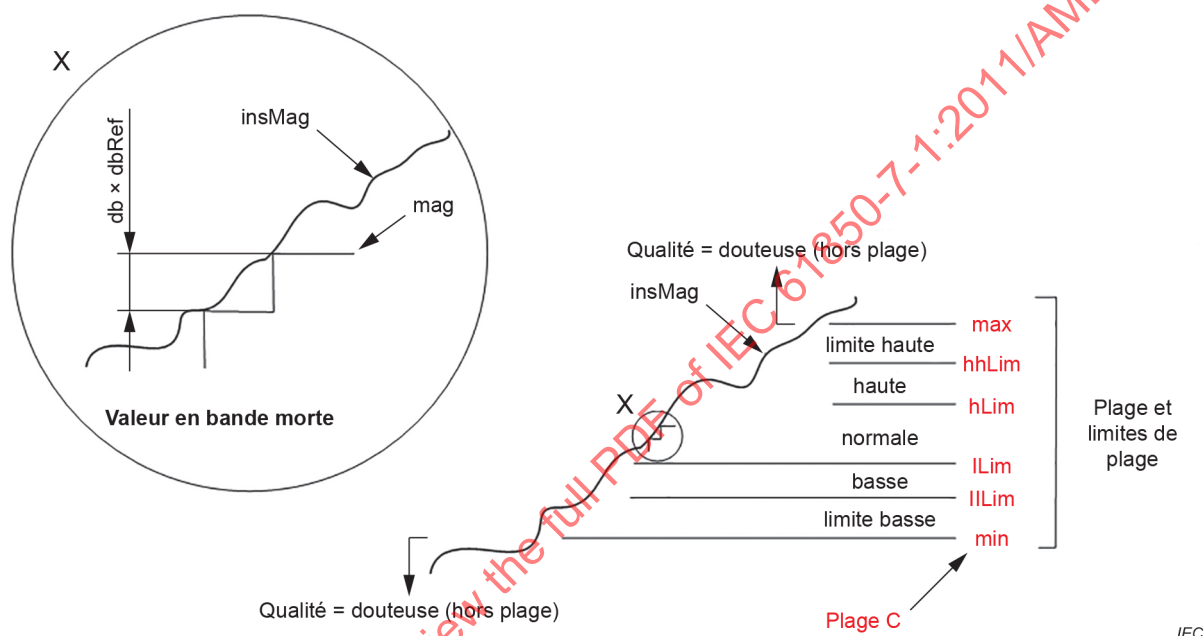
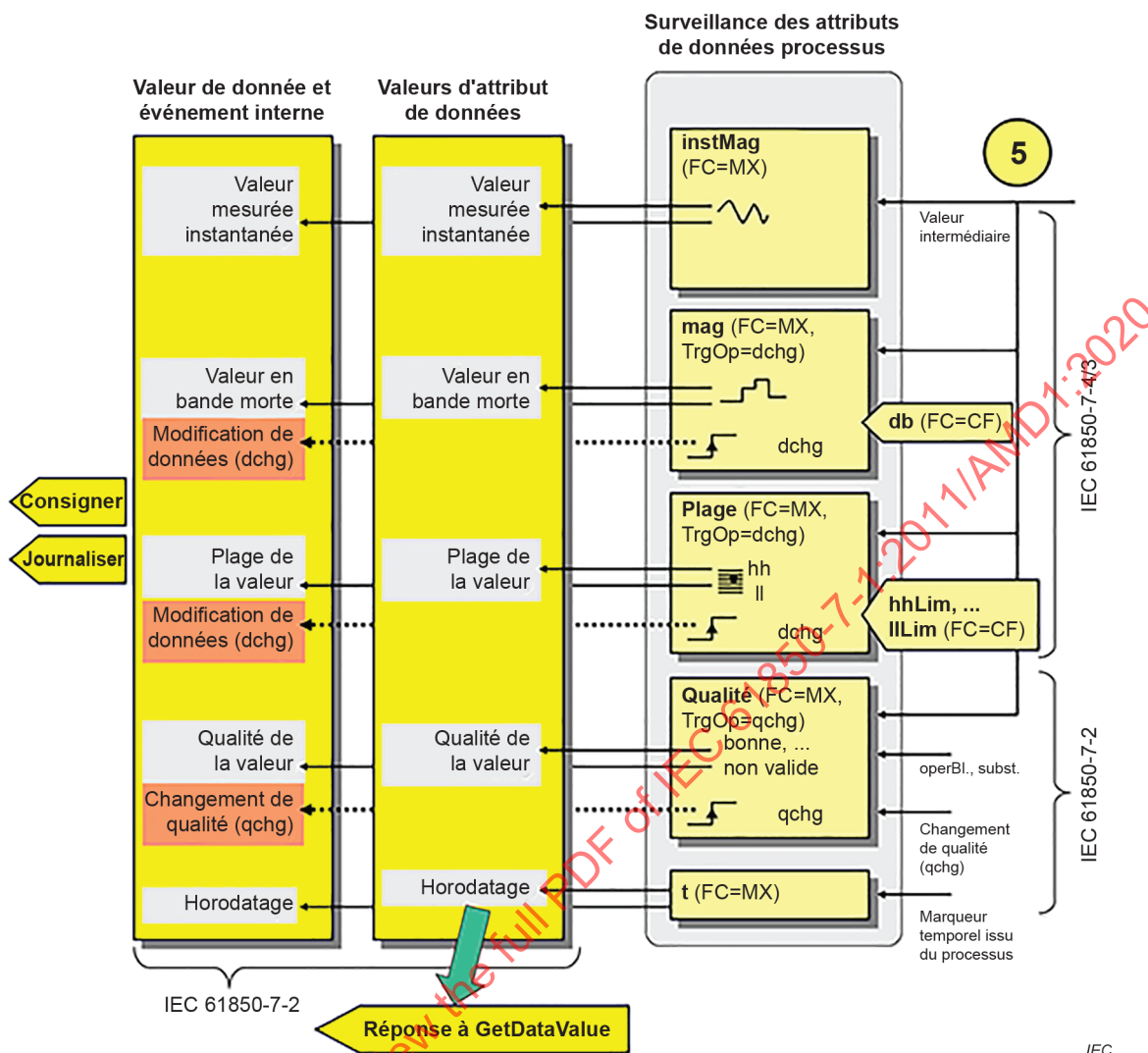


Figure 19 – Plage et valeur en bande morte (à titre conceptuel)

La valeur de la configuration de bande morte db doit représenter le pourcentage de la référence de bande morte dbRef en unités de 0,001% (bande morte fixe). Si dbRef a la valeur 0, cela signifie que la valeur de db doit représenter le pourcentage des dernières valeurs transmises en unités de 0,001 %. Cela permet un calcul de bande morte en fonction de la dernière valeur actualisée (bande morte variable).

Figure 20 – Modèle d'entrée pour valeurs analogiques (étape 2) (à titre conceptuel)

Remplacer la Figure 20 existante par la nouvelle figure suivante:



Supprimer le quatrième alinéa sous la Figure 20.

6.4.3.3.2 Rapports relatifs aux données

Remplacer le Tableau 4 existant par le nouveau tableau suivant:

Tableau 4 – Comparaison de méthodes d'accès aux données

Méthode de récupération	Echange d'informations à durée critique	Pouvant perdre des changements (de séquence)	Plusieurs clients devant recevoir les informations	Dernière modification des données mémorisée par	Client type (mais pas exclusif)
Polling (GetDataValues)	NON	OUI	OUI	–	Navigateur
Unbuffered Reporting	OUI	OUI	NON	–	Interface graphique (GUI) en temps réel
Buffered Reporting	OUI	NON	NON	Serveur	Concentrateur de données
Log (utilisé pour la journalisation de SOE)	NON	NON	OUI	Serveur	Postes d'ingénierie

Remplacer la dernière phrase du deuxième alinéa sous le Tableau 4 par la nouvelle phrase suivante:

La production de rapports destinés au client démarre par la mise à "TRUE" de l'attribut "Enable buffer" (activer le tampon); la mise sur "FALSE" arrête la production de rapports.

Remplacer le paragraphe existante avant la Figure 23 par le nouveau texte suivant:

Les mécanismes de base des rapports mis en mémoire tampon sont illustrés à la Figure 23. Les rapports mis ou non mis en mémoire tampon démarrent avec une réservation du bloc de contrôle par le client, suivie d'un re-paramétrage du bloc de contrôle qui consiste en une vérification des paramètres actuels par rapport aux paramètres souhaités. Les rapports seront envoyés au client après que celui-ci active la transmission via le service SetBRCBValues(ReportEnable=TRUE). La transmission des rapports sera interrompue via le service SetBRCBValues(ReportEnable=FALSE).

Avant la dernière phrase du premier alinéa sous la Figure 23, ajouter la nouvelle phrase suivante:

La mise en mémoire tampon est indépendante de l'activation ou non de la production de rapports.

6.4.3.4 Publication de valeurs de données d'homologue à homologue

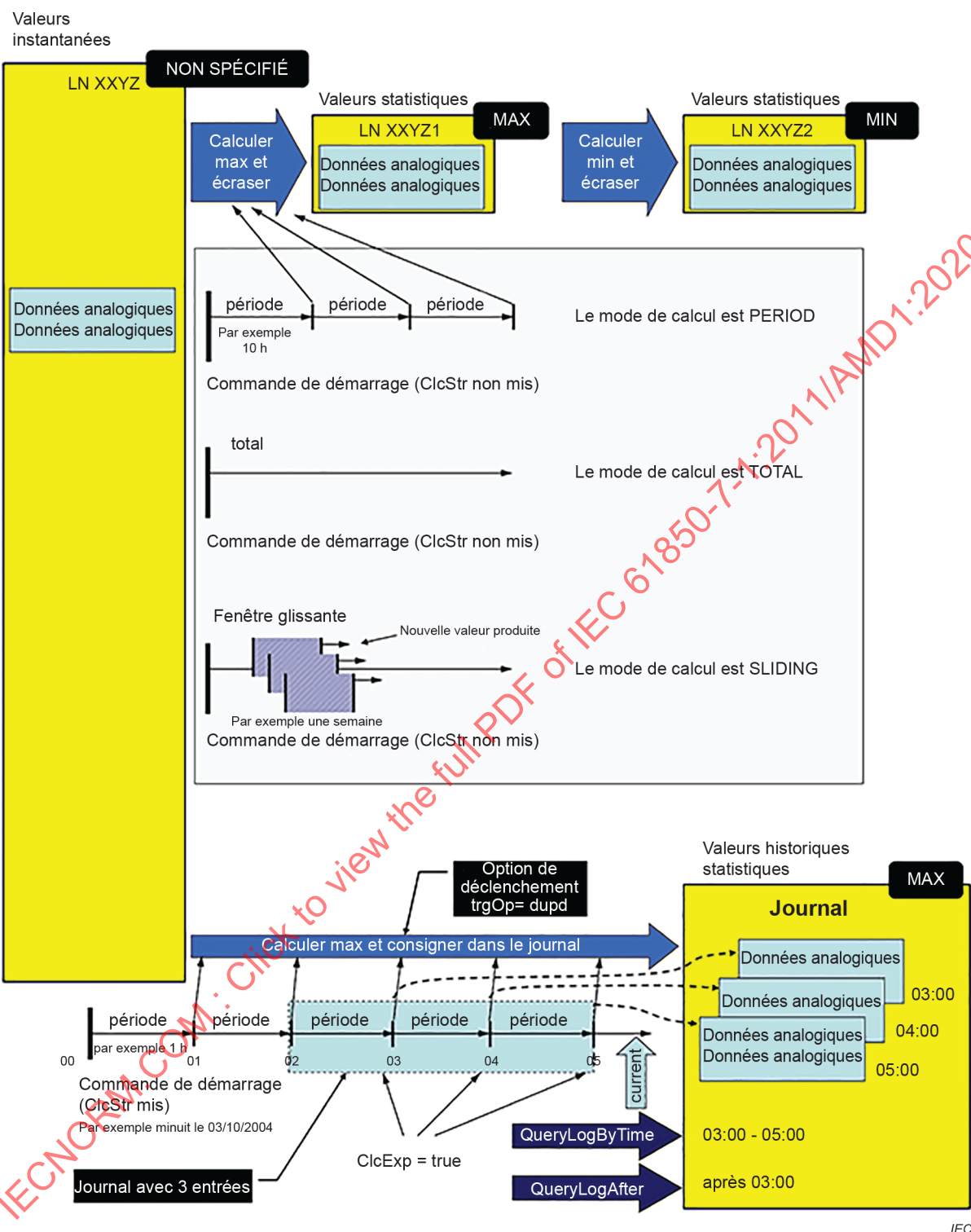
Remplacer le dernier alinéa du Paragraphe 6.4.3.4 par le nouveau texte suivant:

L'événement qui déclenche la publication des valeurs (ainsi que la fréquence et la vitesse à laquelle doivent être publiées ces valeurs) dépend des exigences de la fonction en cours de mise en œuvre.

6.4.4 Modèle pour données statistiques et statistiques historiques

Figure 28 – Modèle conceptuel de données statistiques et statistiques historiques (1)

Remplacer la Figure 28 existante par la nouvelle figure suivante:



IEC

Remplacer l'avant-dernier alinéa dessus la Figure 29 par le nouveau texte suivant:

La partie inférieure de la figure montre le modèle conceptuel des données statistiques historiques. Dans ce modèle, les valeurs sont calculées de façon périodique (dans ce cas, les valeurs maximales avec un mode de calcul mis à PERIOD) et sont stockées en séquence dans un journal. Le calcul de l'exemple démarre à minuit le 03/10/2004. L'intervalle est de 1 h. Après cette première heure, la première entrée de journal est inscrite. Après la deuxième heure, la deuxième entrée contient la valeur de la deuxième heure. Après 5 h, le journal contient les valeurs des trois dernières heures (intervalles 02-03, 03-04, 04-05). L'expiration du calcul statistique est indiquée par l'événement ClcExp mis à " TRUE" et à ce moment, les

valeurs statistiques comportant l'option de déclenchement dupd sont consignées, même si les valeurs n'ont pas été modifiées. Le même concept peut s'appliquer à la production de rapports sur les données statistiques.

Supprimer le dernier alinéa du Paragraphe 6.4.4.

6.4.5 Modèle pour les fonctions de système

6.4.5.1 Généralités

Supprimer les deux dernières phrases du 6.4.5.1.

6.4.5.5 Surveillance de GOOSE et SMV

Ajouter le nouveau texte suivant à la fin du 6.4.5.5:

L'Annexe G normative décrit le flux de configuration en ce qui concerne ces nœuds logiques de surveillance.

7 Vue d'application

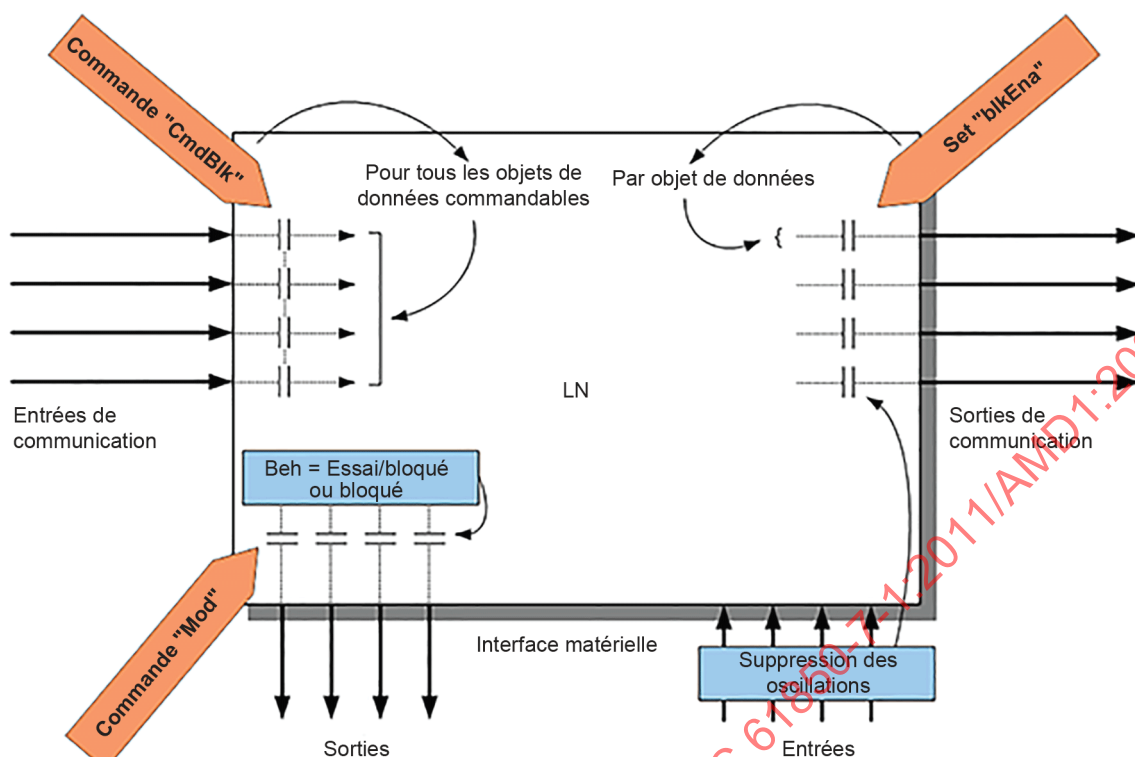
7.3 Mode et comportement d'un nœud logique

Remplacer "bloqué sur marche" par "bloqué" dans le premier alinéa sous la Figure 37.

7.7 Données utilisées pour le blocage des entrées/sorties de nœud logique (blocage opérationnel)

Figure 39 – Données utilisées pour le blocage des entrées/sorties de nœud logique (IEC 61850-7-4)

Remplacer la Figure 39 existante par la nouvelle figure suivante:



IEC

7.7.2 Blocage des commandes entrantes

A la fin du dernier alinéa du 7.7.2, ajouter le nouveau texte suivant:

ou la donnée CmdBlk elle-même.

7.7.3 Blocage des sorties de processus

Remplacer le texte existant du 7.7.3 par le nouveau texte suivant:

La donnée Mod (Mode) est utilisée pour placer un LN dans des modes différents. Le mode test/blocked (Beh = test/blocked or Beh=blocked) doit être interprété comme le mode permettant de désactiver de manière explicite toutes sorties physiques ayant une influence sur le processus. Par exemple, lorsque XCBR.Beh = test/blocked ou blocked, les sorties physiques de fermeture/ouverture du disjoncteur sont bloquées.

7.7.4 Blocage des entrées battantes

Remplacer le titre du 7.7.4 par le nouveau texte suivant:

Blocage de la communication des mises à jour de sorties de statut

Remplacer la première phrase du paragraphe par le nouveau texte suivant:

Un opérateur ou une fonction automatique peut bloquer la mise à jour de la valeur des sorties de communication (les mises à jour liées aux entrées battantes, par exemple).

7.8 Données utilisées pour les essais

7.8.2 Signaux de multidiffusion utilisés pour la simulation

Remplacer le troisième alinéa du Paragraphe 7.8.2 par le nouveau texte suivant:

Pour permettre à l'IED1 de traiter le message GOOSE 1 simulé au lieu du message GOOSE réel, les données Sim.stVal dans le nœud logique LPHD1 doivent avoir la valeur " TRUE". Le processus du message simulé prend effet uniquement à la réception du premier message simulé. Cela dure tant que Sim.stVal ne reprend pas la valeur " FALSE". Le traitement du message simulé est assuré selon GOOSE/SMV. Autrement dit, les autres signaux, GOOSE 2 et GOOSE 3 sont normalement traités par l'IED1. Noter que la commutation entre les signaux normaux et les signaux simulés doit être assurée pour l'ensemble de l'IED. Dans l'exemple, cela signifie que si un nouveau signal GOOSE 2 simulé apparaît, il remplace le signal GOOSE 2 réel. L'exemple précédent n'implique aucune séquence spécifique en ce qui concerne la connexion du dispositif de simulation et le réglage de la donnée Sim.stVal sur "TRUE". Différentes séquences peuvent être utilisées. Un diagramme d'états peut néanmoins être consulté à la Figure 80 et à la Figure 81. Il décrit les différents états impliqués et représente la surveillance d'un abonnement GOOSE/SV en ce qui concerne l'état d'abonnement actif voire l'abonnement au flux simulé. Les objets de données concernés sont définis dans les nœuds logiques LGOS et LSVS de l'IEC 61850-7-4.

Ajouter, à la fin du troisième alinéa du 7.8.2, les nouvelles Figures 80 et 81:

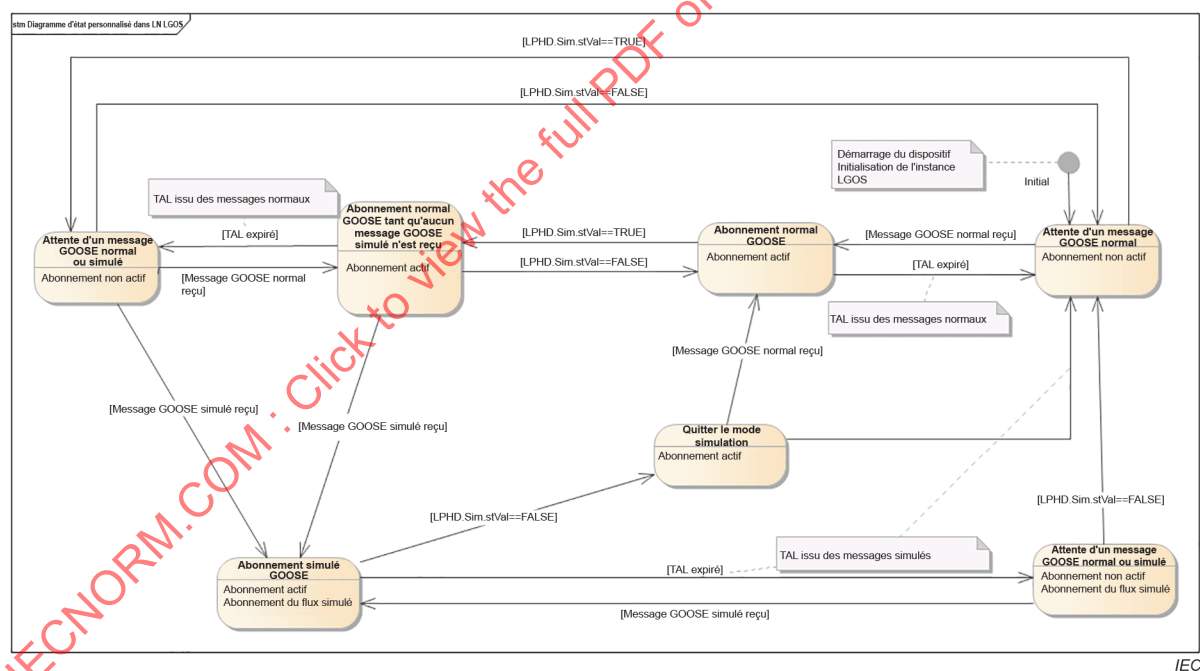


Figure 80 – Diagramme d'états de surveillance d'un abonnement GOOSE

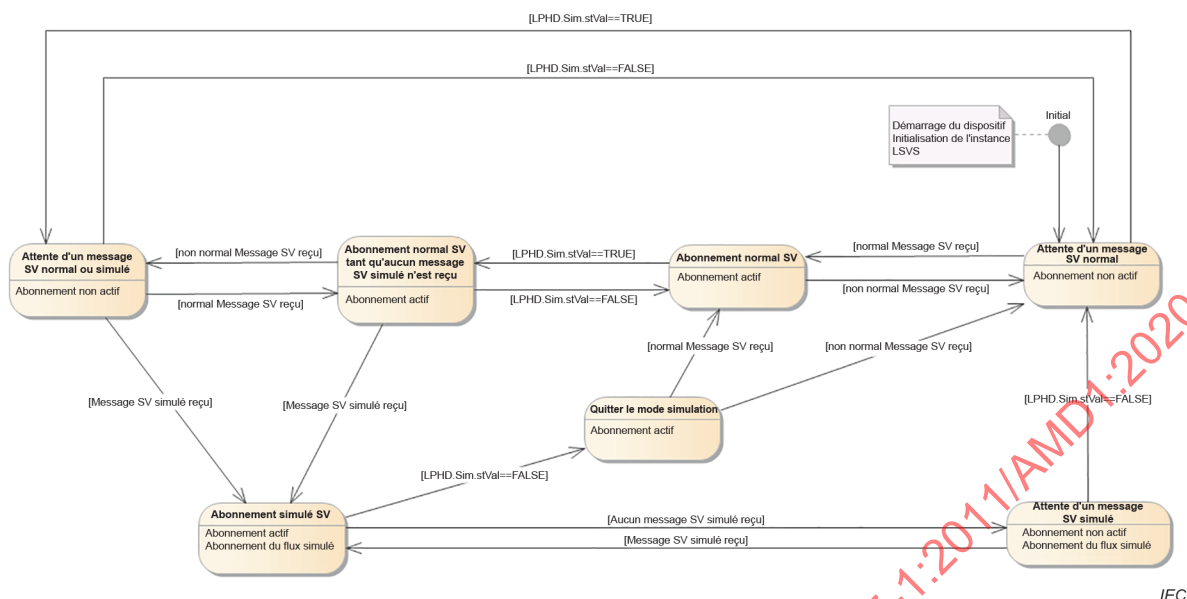
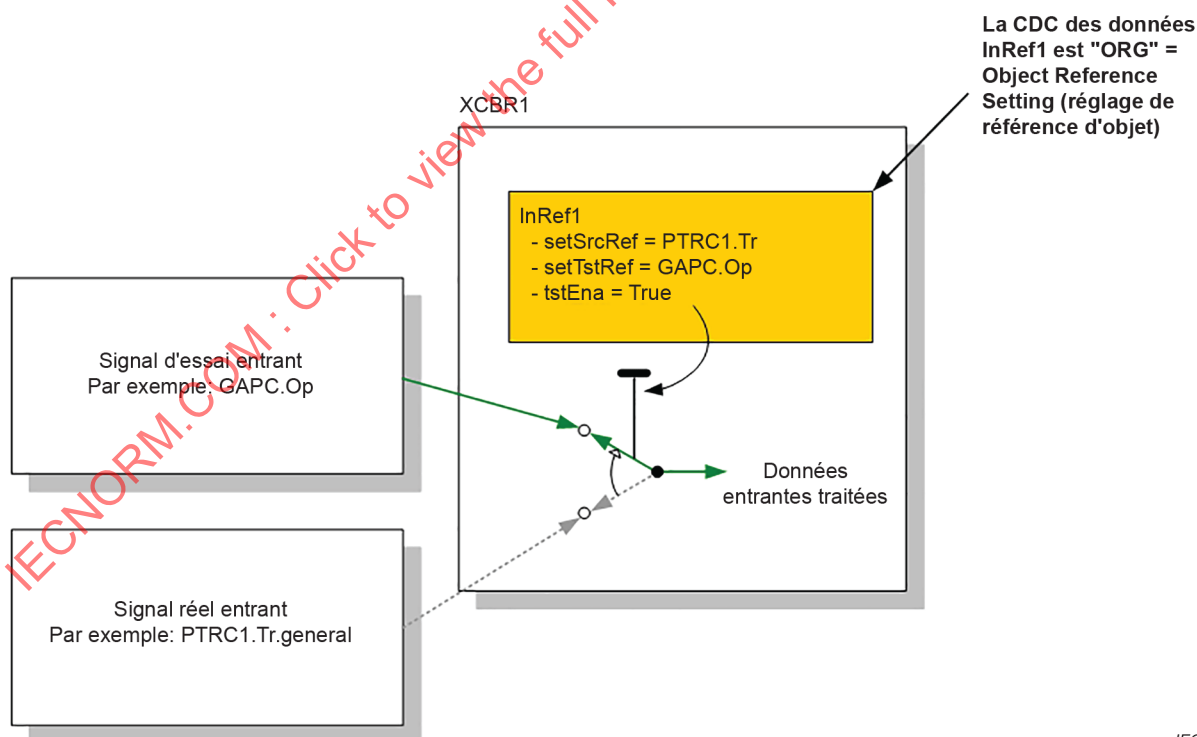


Figure 81 – Diagramme d'états de surveillance d'un abonnement SV

7.8.3 Signaux d'entrée utilisés pour les essais

Figure 41 – Exemple de signaux d'entrée utilisés pour les essais

Remplacer la Figure 41 existante par la nouvelle figure suivante:



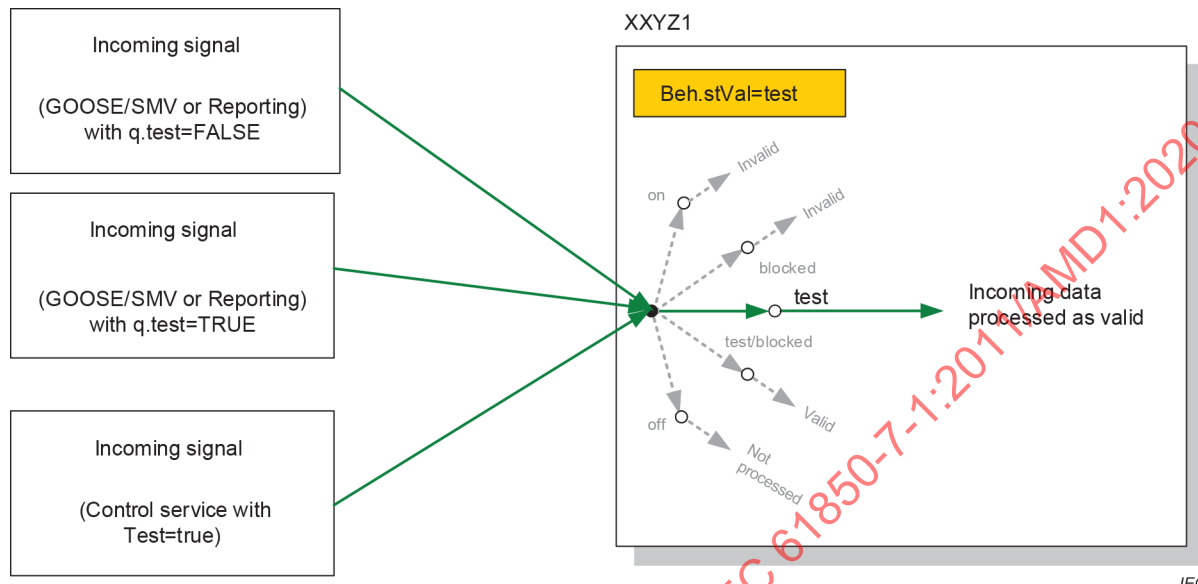
Ajouter le nouveau texte suivant à la fin du dernier alinéa du 7.8.3:

Le signal spécifié dans setTstRef et le bloc de contrôle associé srcTstCB doivent être cohérents et se rapporter à un bloc de contrôle entièrement configuré (c'est-à-dire incluant la configuration de communication, le cas échéant) dans le fichier SCD.

7.8.4 Mode d'essai

Figure 42 – Exemple de mode d'essai

Remplacer la Figure 42 existante par la nouvelle figure suivante:



Remplacer le dernier alinéa du 7.8.4 par le nouveau texte suivant:

Les exigences complètes relatives aux différents états de Mod/Beh peuvent être consultées dans l'IEC 61850-7-4.

8 Vue de dispositif

8.2 Deuxième étape de modélisation – modèle de dispositif logique

8.2.1 Concept de dispositif logique

Remplacer le deuxième alinéa sous la Figure 44 par le nouveau texte suivant:

Dans l'exemple de la Figure 45, le dispositif logique "LD1" contient trois nœuds logiques: LLN0, LPHD et LN, ce dernier représentant un LN spécifique à l'application, comme défini dans la série IEC 61850-7-x. Le nœud logique zéro (LLN0) représente les données communes du dispositif logique. Par exemple, le mode LLN0 est utilisé pour commander le mode du dispositif logique tout entier et, par conséquent, le mode de chaque nœud logique faisant partie du dispositif logique. Le nœud logique dispositif physique (LPHD) représente les données communes du dispositif physique hébergeant le dispositif logique. LLN0 doit être défini dans chaque dispositif logique alors que LPHD doit être défini dans au moins un dispositif logique. De nombreux autres nœuds logiques du système (Groupe L) appartiennent à l'IED (LTRK, par exemple), ce qui permet de suivre les paramètres de service d'un serveur à l'intérieur de l'IED ou de LTIM qui sert à différentes configurations concernant l'heure locale d'un IED. La Figure 45 donne un exemple de bonne pratique. Le dispositif logique appelé "SYS" regroupe tous les nœuds logiques du système appartenant au même IED. Ces nœuds logiques du système peuvent également être répartis dans une hiérarchie de dispositifs logiques (voir 8.2.5).

8.2.3 Passerelles et proxys

Remplacer le texte existant du 8.2.3, y compris la Figure 47, par le nouveau texte et les nouvelles figures suivants:

Les passerelles sont des dispositifs d'interconnexion de réseaux qui convertissent des protocoles en d'autres protocoles. Par exemple, des passerelles peuvent convertir des données non IEC 61850 en données IEC 61850.

Les proxys sont des dispositifs spéciaux qui miroitent des dispositifs logiques situés dans d'autres dispositifs physiques conformes à l'IEC 61850. Par conséquent, ces dispositifs logiques sont transparents d'un point de vue fonctionnel. Ils peuvent être identifiés indépendamment de leur emplacement (dans un dispositif séparé relié au réseau ou dans un dispositif proxy).

Les dispositifs logiques permettent la construction de proxys et/ou de passerelles.

La Figure 47 montre comment un dispositif physique est associé par mapping à un dispositif agissant comme un proxy et/ou une passerelle. Dans la figure, LN se réfère à un LN spécifique à l'application, comme défini dans la série IEC 61850-7-x.

Le dispositif logique IED1LD1 est "copié" sur le proxy/la passerelle. Le LPHD de IED1LD1 dans le proxy/la passerelle représente le dispositif physique IED1.

Les dispositifs logiques qui miroitent des dispositifs logiques d'autres dispositifs physiques doivent fournir un LPHD qui représente le dispositif physique distant sur lequel réside le LD d'origine (par exemple IED1LD1). Ces dispositifs logiques doivent avoir la donnée LPHD.Proxy.stVal du LPHD mise à "TRUE".

Les dispositifs logiques qui ne miroitent pas des dispositifs logiques d'autres dispositifs physiques doivent fournir un LPHD qui représente le dispositif physique sur lequel ils résident. Dans la Figure 47, ce dispositif logique est PXY_GTW1LD1. Il doit être mis en œuvre pour représenter les informations relatives au proxy/à la passerelle. Les nœuds logiques LLN0 et LPHD de ce dispositif logique doivent représenter les informations relatives au dispositif proxy/passerelle. Le dispositif logique peut aussi contenir des nœuds logiques spécifiques au domaine.

Dans l'exemple, le dispositif physique IED2 est également utilisé en tant que proxy par le proxy/la passerelle. Le dispositif contient une hiérarchie de dispositifs logiques. Dans ce cas, le LPHD qui représente le dispositif physique distant dans le proxy/la passerelle est uniquement déclaré à la racine de la hiérarchie.

La Figure 47 représente également la manière dont les dispositifs physiques externes non conformes à l'IEC 61850 sont associés par mapping au dispositif proxy/passerelle. IED3LD1 est utilisée pour représenter les informations liées au dispositif physique IED3 non conforme à l'IEC 61850. Le LPHD de IED3LD1 représente le dispositif physique IED3. Les données fournies par IED3 sont associées par mapping à différents nœuds logiques dans IED3LD1.

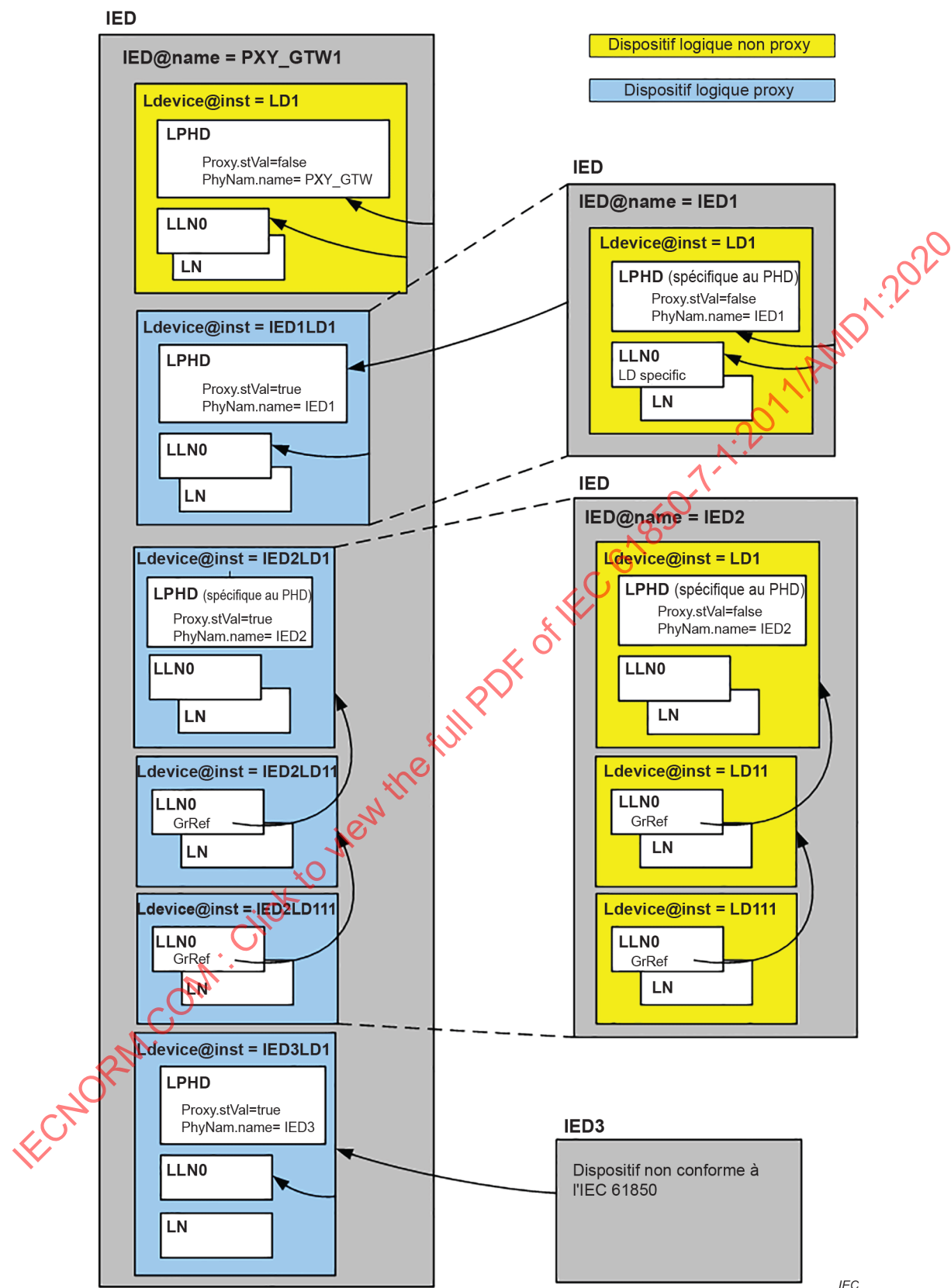


Figure 47 – Dispositifs logiques dans des proxys ou passerelles

La Figure 79 représente le même exemple utilisant la dénomination fonctionnelle des dispositifs logiques. Il peut être noté qu'il est possible d'utiliser les mêmes noms des dispositifs logiques utilisés comme proxys que ceux des dispositifs logiques source, tant que les points d'accès respectifs (serveur du proxy et serveur d'origine) ne sont pas connectés au même réseau: le nom du dispositif logique doit être unique dans la portée d'un réseau.

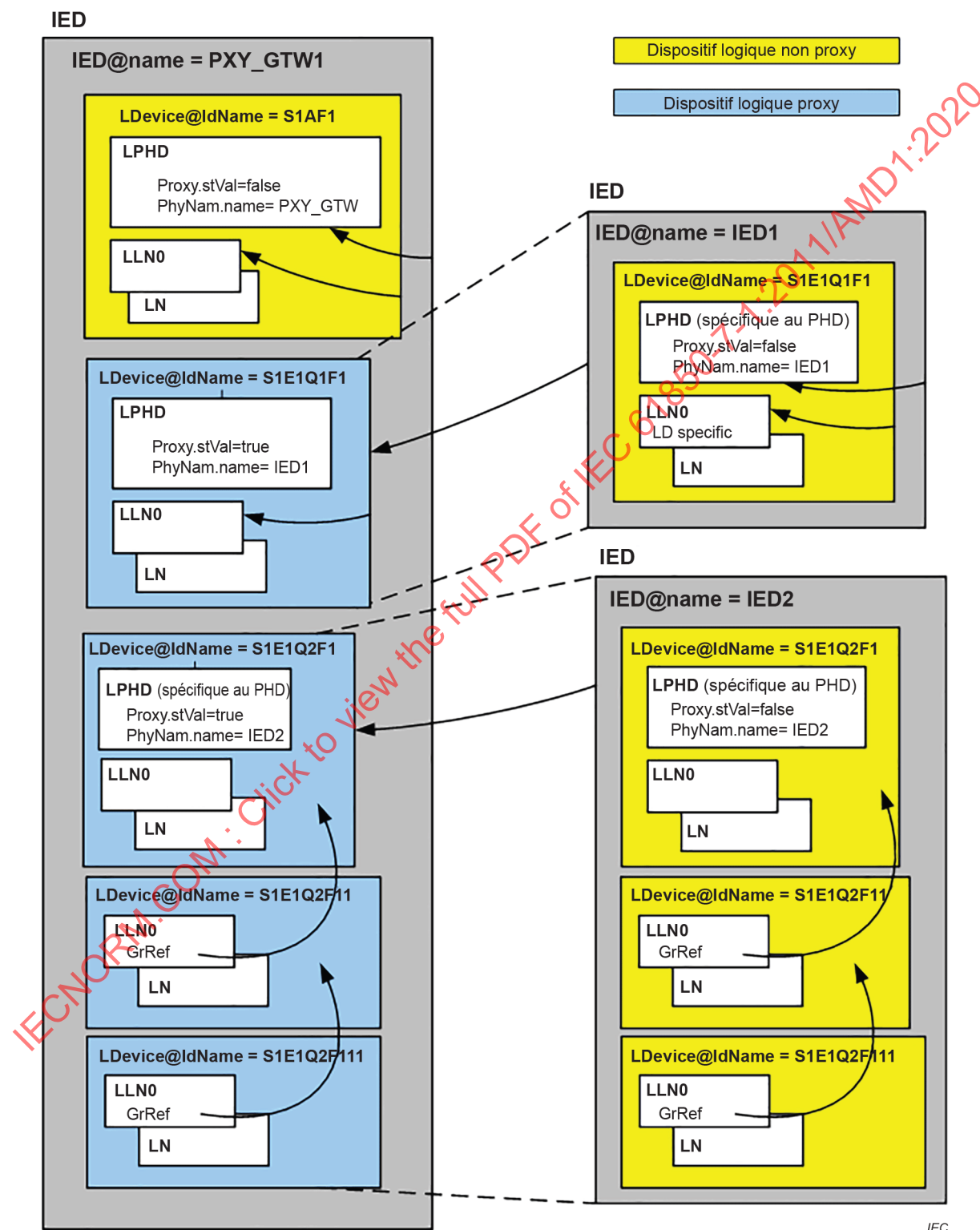
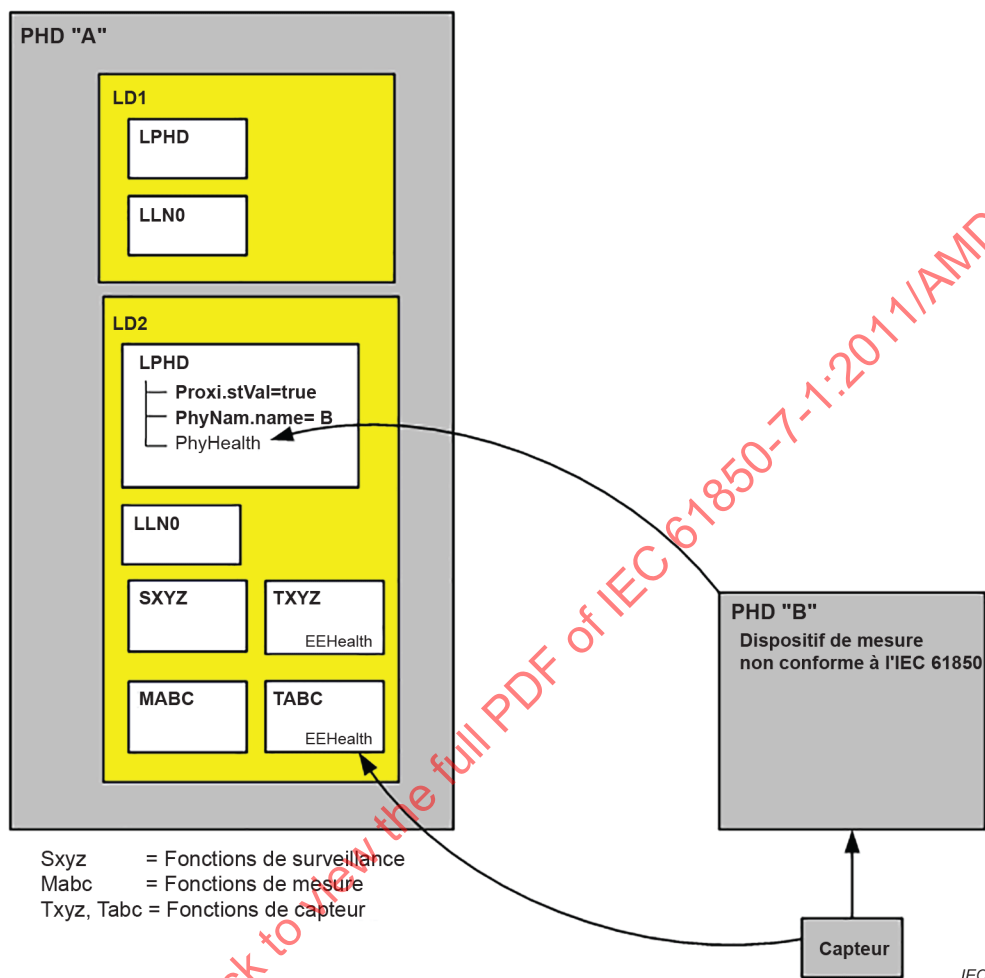


Figure 79 – Dispositifs logiques dans des proxys ou passerelles (dénomination fonctionnelle)

8.2.4 Dispositifs logiques pour surveiller la santé de dispositifs externes

Figure 48 – Dispositifs logiques pour surveiller la santé de dispositifs externes

Remplacer la Figure 48 existante par la nouvelle figure suivante:



8.2.5 Hiérarchie de gestion des dispositifs logiques

Dans le deuxième alinéa du 8.2.5, remplacer "groupe de réglages de référence d'objet" par "réglage de référence d'objet".

Remplacer les troisième et quatrième points de la liste par le nouveau texte suivant:

- la classe de nœuds logiques LPHD doit être présente au niveau de tout dispositif logique racine, c'est-à-dire tout LD où le LLN0 ne contient aucun objet de données GrRef. Elle peut aussi être présente à d'autres niveaux de la hiérarchie,
- un DO du proxy doit être présent uniquement au niveau du dispositif logique racine, s'appliquant à tous les autres niveaux de la hiérarchie associée,

9 Vue de communication

9.2 Modèles de services de la série IEC 61850

Tableau 5 – Modèles et services ACSI

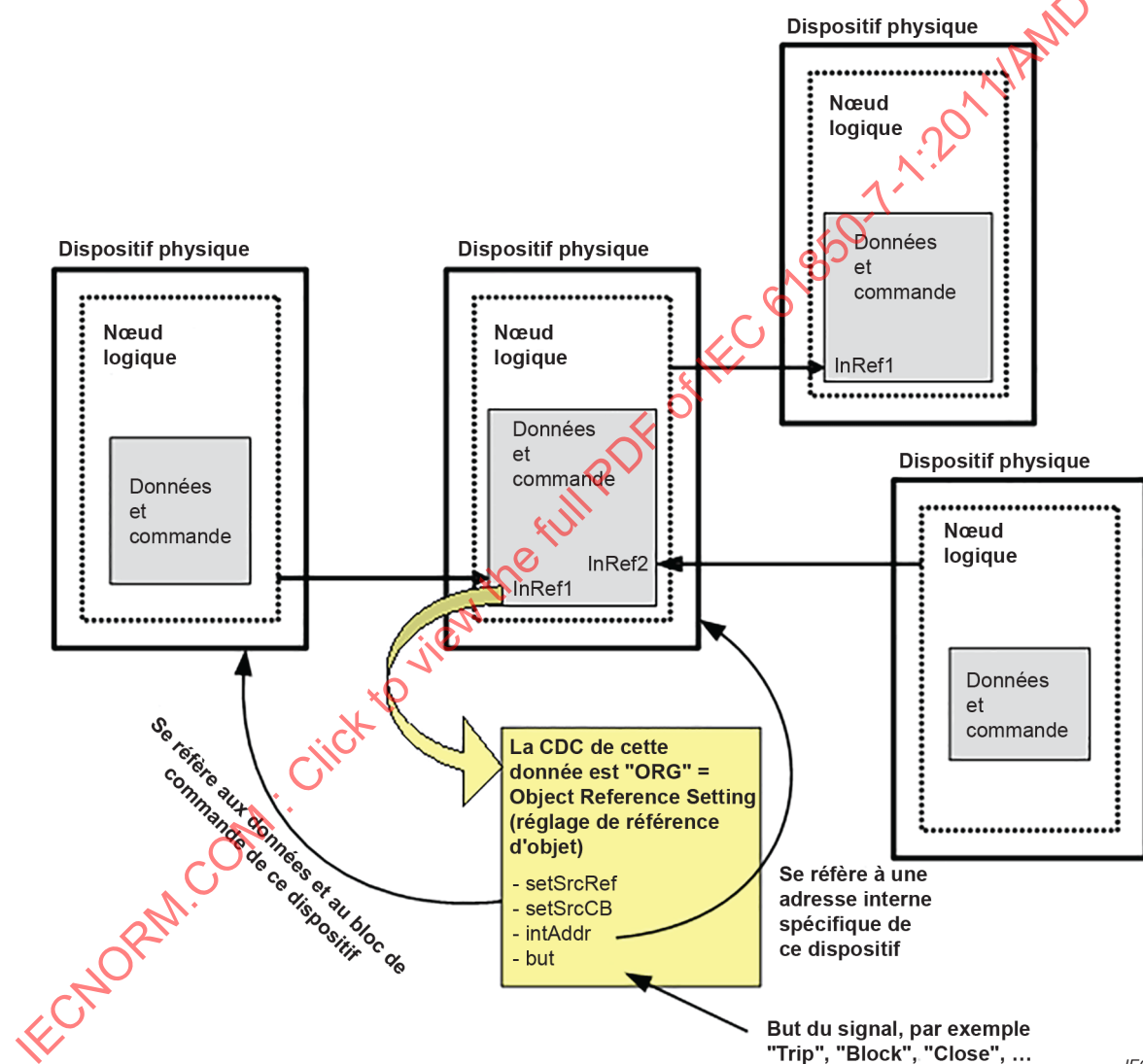
Dans le Tableau 5, remplacer la ligne "Transmission de valeurs échantillonnées" par la nouvelle ligne suivante:

Transmission de valeurs échantillonnées	Transfert rapide et cyclique d'échantillons, par exemple, de transformateurs de mesure	SVC de multidiffusion: SendMSVMessage GetMSVCBValues SetMSVCBValues GetMsvReference GetMSVElementNumber
---	--	--

9.6 Les nœuds logiques communiquent avec des nœuds logiques

Figure 60 – Les nœuds logiques communiquent avec des nœuds logiques

Remplacer la Figure 60 existante par la nouvelle figure suivante:



Remplacer le dernier alinéa du 9.6 par le nouveau texte suivant:

Le langage de configuration SCL (voir IEC 61850-6) permet de configurer le flux de données entre les LN utilisant des éléments "ExtRef" (voir Annexe H).

11 Relations entre l'IEC 61850-7-2, l'IEC 61850-7-3 et l'IEC 61850-7-4

11.2 Exemple 1 – Classe de nœuds logiques et classe d'objets de données

Remplacer le premier alinéa du 11.2 par le nouveau texte suivant:

Le Tableau 6 donne un exemple de liste de classes d'objets de données pour un disjoncteur. Le nom de la classe du disjoncteur est "XCBR". Les classes d'objets de données qui constituent le disjoncteur sont regroupées en quatre catégories (descriptions, informations de statut, commandes et réglages). Chaque catégorie comprend un certain nombre de classes d'objets de données, par exemple, "Mode" et "Switch position". Ces classes d'objets de données sont référencées par leur nom d'objet de données: "Mod" et "Pos". Plus précisément, chaque classe d'objets de données a aussi une classe de données communes définissant les détails, c'est-à-dire les attributs de la classe d'objets de données. La dernière colonne spécifie les conditions de présence pouvant s'appliquer à la classe d'objets de données; autrement dit, elle indique si cette classe d'objets de données est obligatoire (M), facultative (O), interdite dans certains cas (F) ou pouvant être instanciée plusieurs fois (multi). Cette colonne spécifie deux contextes pour les conditions de présence: nds pour le contexte commun du cas Statistiques non dérivées et ds pour le contexte spécifique, où l'instance LN est dérivée d'une autre (Statistiques dérivées). Voir l'IEC 61850-7-4 pour des explications complètes. Ensuite, pour chaque objet de données, deux conditions de présence sont exprimées: un objet dans le cas du contexte nds et un second dans le cas du contexte ds. Par exemple, l'objet de données Pos est obligatoire (M) dans le contexte nds et interdit dans le contexte ds (F).

Remplacer le Tableau 6 existant par le nouveau tableau suivant:

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

Tableau 6 – Nœud logique disjoncteur

Nœud logique: Disjoncteur		Nom: XCBR	
Classe d'objets de données	Nom d'objet de données	Classe de données communes (CDC)	PresConds/ds
Descriptions			
Plaque signalétique du nœud logique.	NamPlt	LPL – Nœud logique plaque signalétique	MONamPlt/ MONamPlt
Plaque signalétique de l'équipement externe (électrique, mécanique ou communication) auquel est associé le nœud logique.	EENam	DPL – Plaque signalétique de dispositif	O/F
Données contrôlables			
(commandable) Mode de fonctionnement du nœud logique de domaine qui peut être modifié par l'opérateur. Le traitement du niveau de qualité ('q') des données reçues est la condition préalable à une interprétation correcte du mode de fonctionnement.	Mod	ENC – Statut entier contrôlable	O/O
(commandable) Position des disjoncteurs/commutateurs.	Pos	DPC – Controllable Double Point	M/F
(commandable) Si l'attribut est mis à "TRUE", cela indique que l'action "open" a été bloquée; peut être défini depuis un autre nœud logique. La capacité de manœuvre ('XCBR.CBOpCap' et 'XSWI.SwOpCap', respectivement pour le disjoncteur et le commutateur) ne reflète pas le blocage d'ouverture.	BlkOpn	SPC – Controllable single point	M/F
(commandable) Si l'attribut est mis à "TRUE", cela indique que l'action "close" a été bloquée; peut être défini depuis un autre nœud logique. La capacité de manœuvre ('XCBR.CBOpCap' et 'XSWI.SwOpCap', respectivement pour le disjoncteur et le commutateur) ne reflète pas le blocage de fermeture.	BlkCls	SPC – Controllable single point	M/F
(commandable) Si l'attribut est mis à "TRUE", cela indique que le moteur de chargeur, utilisé pour prévenir toute surcharge de l'alimentation électrique après un déclenchement de jeu de barres, a été activé.	ChMotEna	SPC – Controllable single point	O/F
(commandable) Si l'attribut est mis à "TRUE", l'autorité de commande se fait au niveau poste et la commande à distance est désactivée; sinon, la commande à distance est admise.	LocSta	SPC – Controllable single point	O/F
Informations de statut			
Valeur en lecture seule, décrivant le comportement d'un nœud logique de domaine. Elle dépend du mode de fonctionnement du courant du nœud logique ('DomainLN.Mod'), ainsi que du mode de fonctionnement du courant du dispositif logique qui le contient ('LLN0.Mod'). Le traitement du niveau de qualité ('q') des données reçues est la condition préalable à une interprétation correcte de 'DomainLN.Beh'.	Beh	ENS – Statut entier	M/M
Reflète l'état des matériels et logiciels liés au nœud logique. Informations plus complètes relatives à la source du problème pouvant être données par un attribut spécifique du nœud logique.	Health	ENS – Statut entier	O/O

Nœud logique: Disjoncteur		Nom: XCBR	
Classe d'objets de données	Nom d'objet de données	Classe de données communes (CDC)	PresConds/ds
Si l'attribut est mis à "TRUE", le comportement de commande est admis à ce niveau.	Loc	SPS – Single point status	M/F
Etat de l'équipement externe (électrique, mécanique ou communication) auquel est associé le nœud logique.	EEHealth	ENS – Statut entier	O/F
Décompte des manœuvres; non réarmables à distance, mais peut être réarmé en local.	OpCnt	INS – Statut entier	M/O
Si l'attribut est mis à "TRUE", le fonctionnement (distant) a été basculé en local. Cette commutation est toujours effectuée en local avec une clé physique ou un commutateur à bascule qui peut comporter un jeu de contacts à partir desquels la position peut être lue.	LocKey	SPS – Single point status	O/F
Capacité physique du disjoncteur à fonctionner, qui reflète l'énergie disponible dans le système d'entraînement.	CBOPCap	ENS – Statut entier	O/F
Capacité de coupure en un point de l'onde	POWCap	ENS – Statut entier	O/F
Capacité de manœuvre disponible lorsque le mécanisme de commutation est à pleine charge, exprimée en tant que nombre maximal de cycles de fermeture-ouverture à effectuer. Par exemple, lorsqu'un cycle de fermeture-ouverture peut être effectué, 'MaxOpCap'=1.	MaxOpCap	ENS – Statut entier	O/O
Informations relatives aux réglages			
Temps de fermeture du disjoncteur. Le temps est utilisé pour compenser le temps de fermeture du disjoncteur, c'est-à-dire que la commande de fermeture bénéficie d'un certain temps (défini) avant la concordance des phases. Il s'agit d'une propriété du disjoncteur qui est sensible au vieillissement.	CBTmms	ING – Réglage du statut entier	O/F

Remplacer le Tableau 7 existant par le nouveau tableau suivant:

Tableau 7 – Controllable double point (DPC)

cdclId = DPC, nom de classe UML = DPC					
Nom d'attribut	Type d'attribut	FC	TrgOp	(Valeur/plage de valeurs) Explication	PresCond
DataAttribute pour statut					
origin	Originator	ST		Voir 'SPC.origin'.	O
ctlNum	INT8U	ST		Voir 'SPC.ctlNum'.	O
stVal	DpStatusKind	ST	dchg	Valeur de statut de l'objet de données contrôlable.	M
q	Quality	ST	qchg	Qualité de la valeur dans 'stVal'.	M
t	Timestamp	ST		Horodatage de la dernière modification de la valeur de 'stVal' ou 'q'.	M
stSeld	BOOLEAN	ST	dchg	Voir 'SPC.stSeld'.	MOsbo
DataAttribute pour miroir de commande					
opRcvd	BOOLEAN	OR	dchg	hérité de: ControlTestingCDC	O
opOk	BOOLEAN	OR	dchg	hérité de: ControlTestingCDC	O
tOpOk	Timestamp	OR		hérité de: ControlTestingCDC	O
DataAttribute pour substitution et bloqué					
subEna	BOOLEAN	SV		hérité de: SubstitutionCDC	MFsubst
subVal	DpStatusKind	SV		Valeur utilisée pour remplacer 'stVal'.	MFsubst
subQ	Quality	SV		hérité de: SubstitutionCDC	MFsubst
subID	VisString64	SV		hérité de: SubstitutionCDC	MFsubst
blkEna	BOOLEAN	BL		hérité de: SubstitutionCDC	O
DataAttribute pour configuration, description et extension					
pulseConfig	PulseConfig	CF	dchg	Voir 'SPC.pulseConfig'.	O
ctlModel	CtlModelKind	CF	dchg	Voir 'SPC.ctlModel'.	M
sboTimeout	INT32U	CF	dchg	Voir 'SPC.sboTimeout'.	MOsbo
sboClass	SboClassKind	CF	dchg	(default=operate-once) Voir 'SPC.sboClass'.	O
operTimeout	INT32U	CF	dchg	Voir 'SPC.operTimeout'.	MOenhanced
d	VisString255	CC		hérité de: BasePrimitiveCDC	O
dU	Unicode255	CC		hérité de: BasePrimitiveCDC	O
cdcName	VisString255	EX		hérité de: BasePrimitiveCDC	O
dataNs	VisString255	EX		hérité de: BasePrimitiveCDC	MOdataNs
Paramètres pour services de commande					
ctlVal	BOOLEAN			Paramètre de service qui détermine l'activité de la commande ('FALSE' pour désactivée, 'TRUE' pour activée).	

Remplacer la Figure 64 existante par la nouvelle figure suivante:

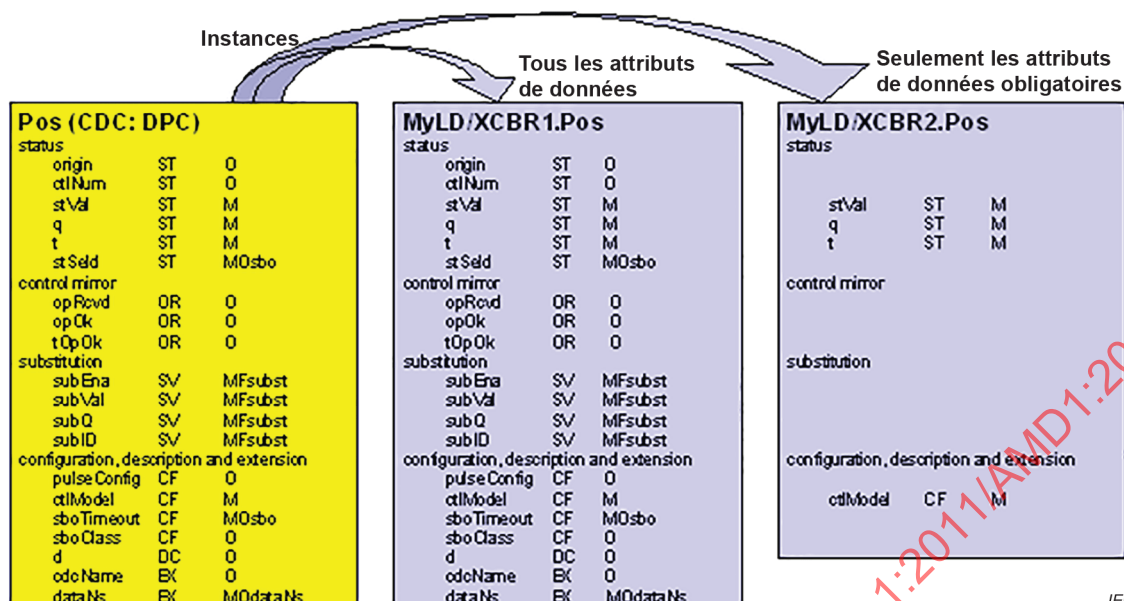


Figure 64 – Instances d'une classe d'objets de données (à titre conceptuel)

Remplacer le deuxième alinéa après la Figure 64 par le nouveau texte suivant:

Les conditions dans la dernière colonne de la classe d'objets de données Pos sont les suivantes (extrait de l'IEC 61850-7-2):

Remplacer le tableau existant qui suit le second alinéa après la Figure 64 par le nouveau tableau suivant:

Abréviation	Condition
M	L'attribut est obligatoire
O	L'attribut est facultatif
MOsbo	L'élément est obligatoire si le modèle de commande déclaré prend en charge 'sbo-with-normal-security' ou 'sbo-with-enhanced-security'; sinon, il est facultatif et la valeur n'a aucun impact.
MFsubst	L'élément est obligatoire si la substitution est prise en charge (pour la substitution, voir l'IEC 61850-7-2); sinon, il est interdit.
MOenhanced	L'élément est obligatoire si le modèle de commande déclaré prend en charge 'direct-with-enhanced-security' ou 'sbo-with-enhanced-security'; sinon, il est facultatif et la valeur n'a aucun impact.
MOdataNs	L'élément est obligatoire si l'espace de noms de son objet de données s'écarte de l'espace de noms de son nœud logique; sinon, il est facultatif. Pour l'utilisation de l'espace de noms, voir l'IEC 61850-7-1.
...	...

12 Méthode de spécification formelle

12.2 Modélisation de classe

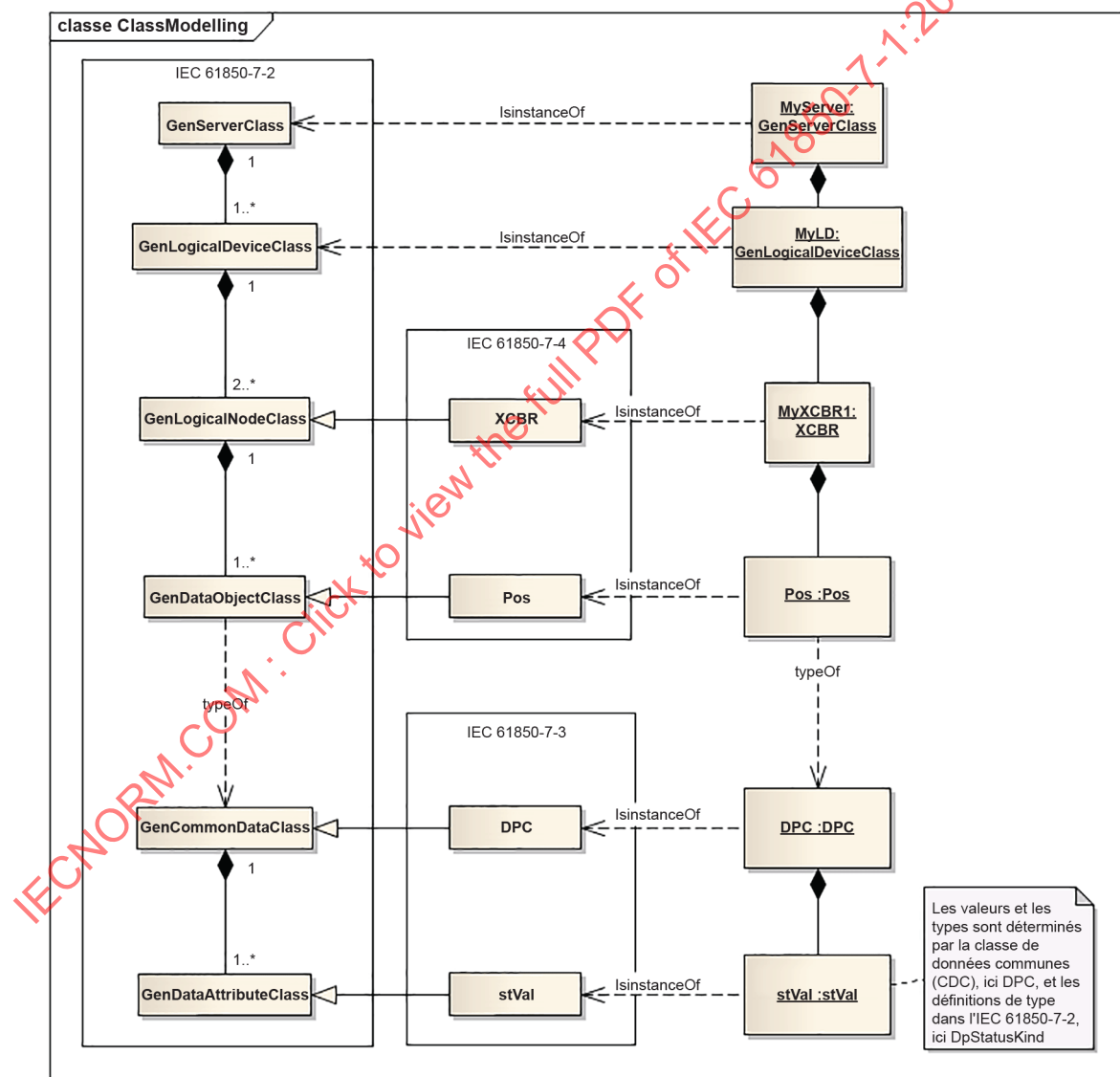
12.2.1 Vue générale

Remplacer le deuxième alinéa du 12.2.1 par le nouveau texte suivant:

La Figure 66 représente le modèle de classe de la série IEC 61850-7-x. Le modèle générique à gauche est défini dans l'IEC 61850-7-2. La classe GenServer, qui représente le comportement externe d'un dispositif, se compose de (1 à n) classes GenLogicalDevice. Les classes GenLogicalDevice comprennent au moins deux classes GenLogicalNode (2 à n).

Figure 66 – Exemple de modèle abstrait de données pour l'IEC 61850-7-x

Remplacer la Figure 66 existante par la nouvelle figure suivante:



12.2.2 Classe de données communes

Tableau 9 – Classe de données communes "Single point status" (SPS)

Remplacer le Tableau 9 existant par le nouveau tableau suivant:

cdclId = SPS, nom de classe UML = SPS					
Nom d'attribut	Type d'attribut	FC	TrgOp	(Valeur/plage de valeurs) Explication	PresCond
DataAttribute pour statut					
stVal	BOOLEAN	ST	dchg	Valeur des données.	M
q	Quality	ST	qchg	Qualité de la valeur dans 'stVal'.	M
t	Timestamp	ST		Horodatage de la dernière modification de la valeur de 'stVal' ou 'q'.	M
DataAttribute pour substitution et bloqué					
subEna	BOOLEAN	SV		hérité de: SubstitutionCDC	MFsubst
subVal	BOOLEAN	SV		Valeur utilisée pour remplacer 'stVal'.	MFsubst
subQ	Quality	SV		hérité de: SubstitutionCDC	MFsubst
subID	VisString64	SV		hérité de: SubstitutionCDC	MFsubst
blkEna	BOOLEAN	BL		hérité de: SubstitutionCDC	O
DataAttribute pour configuration, description et extension					
d	VisString255	CC		hérité de: BasePrimitiveCDC	O
dU	Unicode255	CC		hérité de: BasePrimitiveCDC	O
cdcName	VisString255	EX		hérité de: BasePrimitiveCDC	O
dataNs	VisString255	EX		hérité de: BasePrimitiveCDC	MOdataNs

Remplacer les deuxième et troisième alinéas existants du 12.2.2 par le nouveau texte suivant:

La première colonne représente le nom de l'attribut, la deuxième spécifie le type d'attribut. Un attribut qui est constitué de plusieurs composants est représenté dans l'exemple du Tableau 10 (extrait de la classe Quality de l'IEC 61850-7-2).

Les composants du type Quality (par exemple, validity ou detailQual) sont des composants d'attribut de données. Les types d'attributs des composants d'attribut de données (par exemple, PACKED LIST ou CODED ENUM) sont définis dans l'IEC 61850-7-2. Un extrait de detailQual issu de l'IEC 61850-7-2 est donné dans le Tableau 16.

Tableau 10 – Définition des attributs de composants de Quality

Remplacer le Tableau 10 existant par le nouveau tableau suivant:

atId = Quality			
Nom d'attribut	Type d'attribut	(Valeur/plage de valeurs) Explication	PresCond
validity	ValidityKind	Validité de la valeur, sous forme d'informations condensées destinées au client. Dans le cas où cette valeur n'est pas 'good', les raisons peuvent être consultées dans 'detailQual'.	M
detailQual	DetailQual	Décrit les raisons pour lesquelles 'validity' n'a pas la valeur 'good'.	M
source	SourceKind	(default=process) Définit la source d'une valeur. NOTE 1 La substitution peut être effectuée en local ou par l'intermédiaire des services de communication. Dans le second cas, des attributs spécifiques avec FC=SV sont utilisés. NOTE 2 Il existe différentes méthodes permettant d'effacer une substitution. A titre d'exemple, une substitution qui a été effectuée en suivant une condition non valide peut être effacée automatiquement si la condition non valide est effacée. Il s'agit cependant d'un problème localisé qui ne relève pas du domaine d'application de la présente norme.	M
test	BOOLEAN	(default= FALSE) Si "TRUE", la valeur est une valeur d'essai. Le traitement de la qualité d'essai dans le client doit être tel que décrit dans l'IEC 61850-7-4. Ce bit doit être complètement indépendant des autres bits dans le descripteur de qualité.	M
operatorBlocked	BOOLEAN	(default= FALSE) Si "TRUE", la mise à jour ultérieure de la valeur a été bloquée par un opérateur. La valeur doit correspondre à l'information qui a été obtenue avant le blocage. Si cet indicateur est réglé, 'detailQual.oldData' doit également être réglé. L'opérateur doit utiliser l'attribut de données 'CDC.blkEna' pour bloquer la mise à jour de la valeur. NOTE Un opérateur et une fonction automatique peuvent bloquer la mise à jour des communications ainsi que la mise à jour des entrées. Dans les deux cas, 'detailQual.oldData' est réglé. Si le blocage est effectué par un opérateur, cet indicateur est alors réglé et il est exigé que l'opérateur efface la condition. EXEMPLE Un opérateur peut bloquer la mise à jour d'une entrée pour enregistrer l'ancienne valeur avant de couper l'alimentation auxiliaire.	M

Supprimer l'alinéa qui suit le Tableau 10 et ajouter le nouveau Tableau 16 suivant après le Tableau 10:

Tableau 16 – Attributs de DetailQual

datId = DetailQual			
Nom d'attribut	Type d'attribut	(Valeur/plage de valeurs) Explication	PresCond
overflow	BOOLEAN	(default= FALSE) Si "TRUE", il est impossible de représenter correctement la valeur de l'attribut auquel la qualité a été associée (utilisée pour les informations relatives à la grandeur mesurée uniquement). EXEMPLE Une valeur mesurée peut ne pas être comprise dans la plage qui peut être représentée par le type de données choisi; par exemple, le type de données est un entier non signé de 16 bits et la valeur est supérieure à 65 535.	M
outOfRange	BOOLEAN	(default= FALSE) Si "TRUE", l'attribut auquel la qualité a été associée n'est pas compris dans une plage de valeurs prédéfinie. Le serveur doit déterminer si la validité (validity) doit être réglée sur invalid (non valide) ou questionable (incertaine) (utilisée pour les informations relatives à la grandeur mesurée uniquement). EXEMPLE Une valeur mesurée peut ne pas être comprise dans une plage prédéfinie, mais le type de données choisi peut toujours représenter la valeur; par exemple, le type de données est un entier non signé de 16 bits et la plage prédéfinie est comprise entre 0 et 40 000: si la valeur est comprise entre 40 001 et 65 535, elle est jugée comme étant hors plage.	M
badReference	BOOLEAN	(default= FALSE) Si "TRUE", la valeur peut ne pas être une valeur correcte en raison d'une référence hors étalonnage. Le serveur doit déterminer si la validité (validity) doit être réglée sur invalid (non valide) ou questionable (incertaine) (utilisée pour les informations relatives à la grandeur mesurée et pour les informations relatives au compteur binaire uniquement).	M
oscillatory	BOOLEAN	(default= FALSE) Pour prévenir toute surcharge des voies de communication pilotée par les événements, il est souhaitable de détecter et de supprimer les entrées binaires battantes (modification rapide). Si un signal varie sur une durée déterminée deux fois dans la même direction (de 0 à 1 ou de 1 à 0), il doit alors être défini en tant qu'oscillation et cet attribut doit être réglé sur "TRUE". Si un nombre configuré de variations transitoires est détecté, elles doivent être supprimées. Pendant cette période, 'Quality.validity' doit être réglé sur 'questionable'. Si le signal est toujours à l'état d'oscillation après le nombre défini de variations, la valeur doit conserver l'état dans lequel elle était au moment où cet indicateur a été réglé. Dans ce cas, la valeur de 'Quality.validity' doit passer de 'questionable' à 'invalid' et elle doit être conservée tant que le signal oscille. Si la configuration est telle qu'il convient de supprimer l'ensemble des variations transitoires, 'Quality.validity' doit immédiatement être réglé sur 'invalid' et cet indicateur doit être réglé sur "TRUE" (utilisé pour les informations de statut seulement).	M
Failure	BOOLEAN	(default= FALSE) Si "TRUE", une fonction de surveillance a détecté une défaillance interne ou externe.	M
oldData	BOOLEAN	(default= FALSE) Si "TRUE", aucune mise à jour n'est effectuée sur une certaine période. La valeur peut être une ancienne valeur qui peut avoir varié dans l'intervalle. Cette période spécifique peut être définie par un attribut qui admet le vieillissement. NOTE Les erreurs "Fail silent", pour lesquelles l'équipement arrête d'envoyer des données, entraînent le réglage de cet indicateur sur "TRUE". Dans ce cas, les dernières informations reçues étaient correctes.	M
inconsistent	BOOLEAN	(default= FALSE) Si "TRUE", une fonction d'évaluation a détecté une incohérence.	M
inaccurate	BOOLEAN	(default= FALSE) Si "TRUE", la valeur ne satisfait pas au niveau de précision indiqué pour la source. EXEMPLE La valeur mesurée du facteur de puissance peut être bruyante (inaccurate) lorsque le courant est très faible.	M

Ajouter le nouveau texte suivant après le Tableau 16:

La colonne FC spécifie la contrainte fonctionnelle, le cas échéant. La contrainte fonctionnelle indique les services qui peuvent être utilisés pour accéder aux valeurs des attributs de données. Le Tableau 11 décrit les services qui sont admis pour les informations de statut. Il s'agit d'un extrait de l'IEC 61850-7-2.

Tableau 11 – Modèle d'informations de statut de base (extrait)

Remplacer le Tableau 11 existant par le nouveau tableau suivant:

Contraintes fonctionnelles		
FC	Sémantique	Description (services admis, valeurs initiales, stockage)
ST	Informations de statut	L'attribut de données doit représenter les informations de statut. La valeur initiale doit être extraite du processus. Note de modélisation: Les services ACSI applicables: - GetDataValues - GetDataDefinition - GetDataDirectory - GetDataSetValues - GetAllDataValue peuvent être un DataSetMember d'un DataSet auquel un bloc de contrôle GOOSE, un bloc de contrôle de rapports, un bloc de contrôle de journaux, ou un bloc de contrôle de valeur échantillonnée fait référence.

Remplacer le texte existant entre les Tableaux 11 et 12 par le nouveau texte suivant:

Les services applicables sont énumérés dans la troisième colonne. Pour toutes les données qui héritent d'attributs issus de la classe de données communes SPS (voir Tableau 11), ces services peuvent accéder aux attributs avec FC=**ST**.

Les options de déclenchement TrgOp spécifient les conditions de déclenchement possibles qui peuvent entraîner l'envoi d'un rapport ou la journalisation d'événements. Les procédures de services doivent être celles spécifiées dans le Tableau 12.

Tableau 12 – Option de déclenchement

Remplacer le Tableau 12 existant par le nouveau tableau suivant:

Options de déclenchement		
TrgOp	Sémantique	Description
dchg	data-change	La variation de la valeur de l'attribut de données associé peut déclencher la génération d'un rapport ou d'une entrée de journal.
qchg	quality-change	La variation de la valeur de l'attribut de données de qualité peut déclencher la génération d'un rapport ou d'une entrée de journal.
dupd	data-update	La mise à jour de la valeur de l'attribut de données associé peut déclencher la génération d'un rapport ou d'une entrée de journal. La valeur mise à jour peut être identique à l'ancienne valeur. Par exemple, le gel de la valeur d'un attribut de données gelable qui met à jour la valeur d'un autre attribut de données peut générer la même valeur existante. Autre exemple, la distance par rapport à une défaillance qui persiste, ou le calcul d'une valeur minimale sur une période et cette même valeur minimale atteinte dans deux fenêtres consécutives de calcul statistique.
dchg, dupd	dchg ou dupd	'dchg' ou 'dupd', à choisir au moment de la définition de l'objet de données dans la portée d'un nœud logique, ou au moment de la mise en œuvre.

Remplacer le dernier alinéa du 12.2.2 par le nouveau texte suivant:

La colonne "(Valeur/plage de valeurs) Explication" contient une description de l'attribut et peut préciser certaines valeurs/plages de valeurs autorisées pour l'attribut. La dernière colonne "PresCond" indique les conditions de présence de l'attribut (optional, conditional, ...).

12.2.3 Classe de nœuds logiques

Tableau 13 – Définition de la classe GenLogicalNodeClass

Remplacer le Tableau 13 existant par le nouveau tableau suivant:

GenLogicalNodeClass		
Nom d'attribut	Type d'attribut	Explication
LNName	ObjectName	Nom d'instance qui identifie sans ambiguïté le nœud logique dans la portée d'un dispositif logique.
LNRef	ObjectReference	Nom de chemin unique du nœud logique: LDName/LNName .
DataObject [1..n]	GenDataObjectClass	Ensemble des objets de données contenus dans le nœud logique. NOTE L'IEC 61850-7-4 définit les classes d'objets de données normalisées.
DataSet [0..n]	DATA-SET	Ensemble des jeux de données contenus dans le nœud logique.
BufferedReportControlBlock [0..n]	BRCB	Ensemble des blocs de contrôle de rapports avec mise en mémoire tampon contenus dans le nœud logique.
UnbufferedReportControlBlock [0..n]	URCB	Ensemble des blocs de contrôle de rapports sans mise en mémoire tampon contenus dans le nœud logique.
LogControlBlock [0..n]	LCB	Ensemble des blocs de contrôle de journaux contenus dans le nœud logique.
SI la classe de LN compatible est égale au LLN0 de l'espace de noms défini dans l'IEC 61850-7-4		
SettingGroupControlBlock [0..1]	SGCB	Bloc de contrôle du groupe de réglages contenu dans le nœud logique.
Log [0..n]	LOG	Ensemble des journaux contenus dans le nœud logique.
GOOSEControlBlock [0..n]	GoCB	Ensemble des blocs de contrôle GOOSE contenus dans le nœud logique.
MulticastSampledValueControlBlock [0..n]	MSVCB	Ensemble des blocs de contrôle des valeurs échantillonnées de multidiffusion contenus dans le nœud logique.
UnicastSampledValueControlBlock [0..n]	USVCB	Ensemble des blocs de contrôle des valeurs échantillonnées de monodiffusion contenus dans le nœud logique.
Services		
GetLogicalNodeDirectory GetAllDataValues		
NOTE L'IEC 61850-7-4 définit les classes de nœuds logiques spécialisées, c'est-à-dire les classes de nœuds logiques compatibles par exemple, XCBR représentant les disjoncteurs. Il est nécessaire que le propriétaire d'espace de noms spécifie de manière explicite la présence du journal dans la définition de ses classes de nœuds logiques compatibles, où la prise en charge du journal est exigée en plus du LLN0 de l'espace de noms défini dans l'IEC 61850-7-4.		

Remplacer les neuvième et dixième alinéas existants du 12.2.3 par le nouveau texte suivant:

La présence des cinq derniers attributs facultatifs n'est disponible que dans la classe de LN LLN0.

Les services qui s'exécutent sur le nœud logique sont les deux services répertoriés à la fin du tableau (GetLogicalNodeDirectory et GetAllDataValues). Toutes les classes utilisées comme des types (répertoriées dans la colonne "Type d'attribut") ont leurs propres services. La classe DATA possède plusieurs services, par exemple GetDataValues et SetDataValues.

12.3 Tableaux des paramètres de service

Remplacer le premier tableau du 12.3 par le nouveau tableau suivant:

Request		
Nom de paramètre	Type de paramètre	Description
Parameter 1...		
Parameter n		
Response+		
Nom de paramètre	Type de paramètre	Description
Parameter 1...		
Parameter n		
Response–		
Nom de paramètre	Type de paramètre	Description
ServiceError		

13 Espaces de noms

Remplacer le texte, les figures et les tableaux existants de l'Article 13 par les nouveaux textes, figures et tableaux suivants:

13.1 Généralités

Chaque classe définie dans l'IEC 61850-7-4 (par exemple LODdouble point status", DPS) et l'IEC 61850-7-2 (par exemple bloc de contrôle de rapports avec mise en mémoire tampon, BRCB) a un nom de classe et une signification spécifique associée à la classe. Pratiquement toutes les classes sont composées d'autres classes. Le modèle de classe hiérarchique de l'IEC 61850-7-x ou d'un autre espace de noms de classe de la série IEC 61850 ou lié à la série IEC 61850 fournit une hiérarchie de dénomination. Chaque nom dans la hiérarchie a une sémantique dans le contexte dans lequel le nom est utilisé.

La Figure 72 fournit un exemple de la définition de noms et leur sémantique dans le contexte d'un poste. L'application devant être modélisée et définie dans la série IEC 61850 peut être le disjoncteur dessiné au milieu à la partie supérieure. Le disjoncteur normalisé est modélisé comme étant un LOGICAL-NODE avec le nom de classe spécifique XCBR. Le disjoncteur fait partie du LOGICAL-DEVICE avec le nom SUBST2. Parmi les autres attributs, le disjoncteur a les informations (modélisées comme classe DATA) qui représentent la position nommée Pos. Parmi les autres informations, la position a un statut (modélisé comme DATA-ATTRIBUTE) nommé stVal. La valeur de statut stVal a quatre valeurs qui représentent le statut possible du disjoncteur réel.

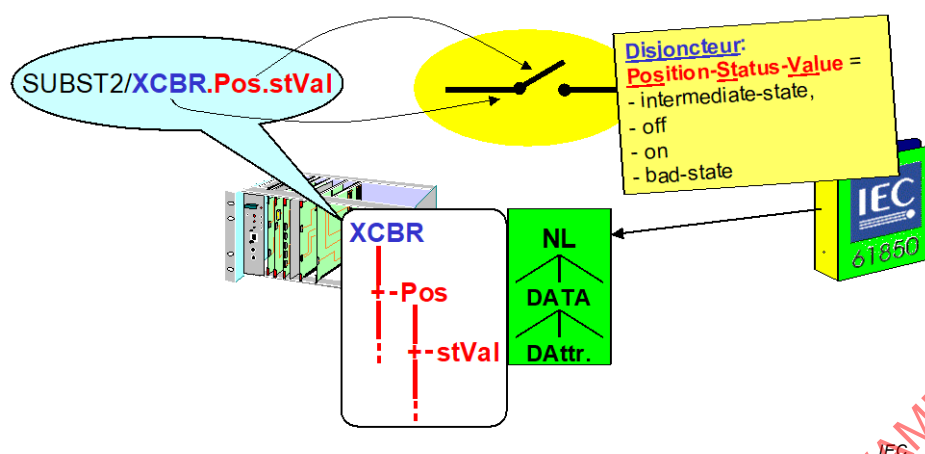


Figure 72 – Définition des noms et sémantique

Si seules les classes définies dans la série IEC 61850-7-x sont utilisées pour bâtir un LOGICAL-DEVICE, la sémantique est celle définie dans la série IEC 61850-7-x.

Pour les applications qui nécessitent des LOGICAL-NODE, DATA ou DATA-ATTRIBUTE, des règles sont fournies pour interpréter sans ambiguïté les noms, c'est-à-dire comprendre le contenu et la signification d'une instance d'une classe dans un contexte spécifique.

L'utilisation d'une référence au contexte défini de la donnée, c'est-à-dire le concept de l'espace de noms, fournit un moyen d'identifier de façon unique la sémantique complète d'une instance d'un LOGICAL-DEVICE, à savoir la sémantique de tous ses LOGICAL-NODES, DATA, DATA-ATTRIBUTES, et toutes les autres instances dans le contexte de son utilisation.

Le concept d'espace de noms permet de différencier les classes définies par différents groupes (tant que les espaces de noms ont des identificateurs uniques).

Toute instance d'une classe des classes définies dans la série IEC 61850 et toutes les instances d'une classe définie comme extension des classes de la série IEC 61850 doivent fournir des informations relatives aux espaces de noms suffisantes pour permettre une interprétation non ambiguë de la sémantique de l'instance. Les instances de classes sont repérées pour identifier l'espace de noms.

13.2 Espaces de noms définis dans la série IEC 61850 ou liés à la série IEC 61850

L'IEC 61850-7-4 définit les espaces de noms LN et DATA pour les classes d'applications génériques. L'IEC 61850-7-3 définit les classes de données communes sur lesquelles reposent les espaces de noms LN et DO. L'IEC 61850-7-2 définit un espace de noms pour les classes (de services) liées aux communications telles que BUFFERED-REPORT-CONTROL-BLOCK, LOG-CONTROL-BLOCK, GenLogicalNodeClass, GenDataObjectClass, DATA-SET et pour les définitions de types tels que les types de base et les types de domaines utilisés dans le modèle de données ou en tant que paramètre de service ACSI.

NOTE L'utilisation mixte de données avec des espaces de noms issus d'autres normes de communication ou issus de définitions privées implique toujours d'adopter la même approche conceptuelle du modèle de données.

Comme représenté à la Figure 73, un espace de noms est, du point de vue conceptuel, un référentiel de classes qui contient différentes classes. Un dispositif logique est constitué d'instances dérivées des classes du référentiel. Le référentiel de classes normalisé fourni dans la série IEC 61850 ou les normes liées à la série IEC 61850, est présenté sur le côté droit de la figure. Un exemple d'espace de noms complémentaire est représenté dans la partie gauche.

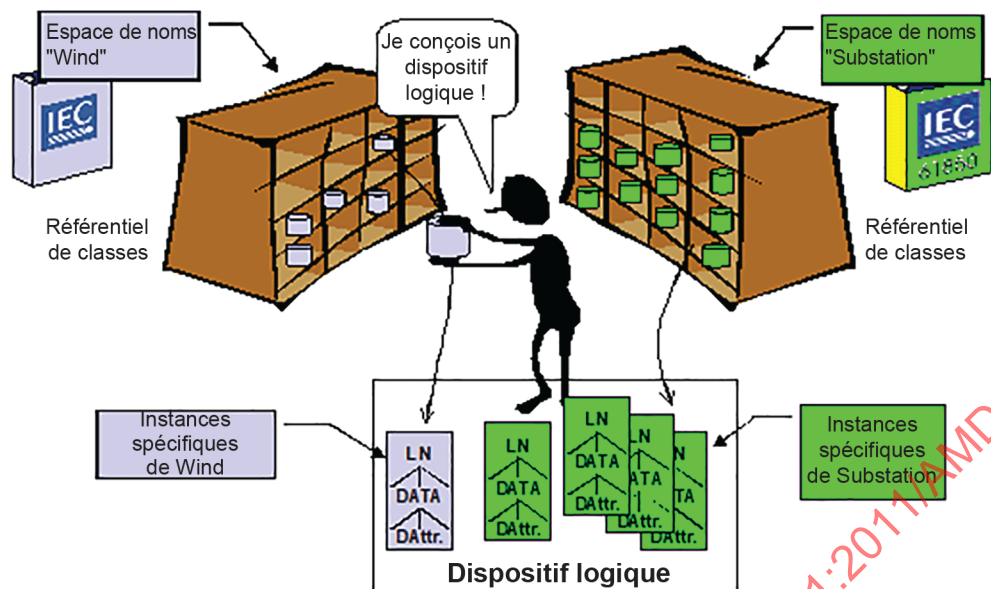


Figure 73 – Espace de noms en tant que référentiel de classes

Les instances qui font partie du dispositif logique sont colorées différemment. Les instances dérivées de la série IEC 61850 sont vertes tandis que les instances dérivées de l'autre norme sont bleues. Comme indiqué à la Figure 74, la désignation de l'espace de noms est représentée par l'attribut "logical device namespace" (espace de noms de dispositif logique):

IdNs = IEC 61850-7-4:2003

Dans cet exemple, l'espace de noms "IEC 61850-7-4:2003" indique que TOUTES les instances au sein de ce dispositif logique sont dérivées des éditions 2003 de l'IEC 61850-7-4, l'IEC 61850-7-3 et l'IEC 61850-7-2. L'espace de noms de dispositif logique peut être interprété comme étant l'espace de noms principal, même s'il n'y a qu'un seul nom. L'attribut IdNs (logical device namespace, espace de noms de dispositif logique) est un attribut contenu dans la plaque signalétique du nœud logique zéro (LLN0). La spécification formelle des espaces de noms peut être consultée en 13.4.

L'IEC 61850-7-4 possède des références normatives qui s'appliquent à l'IEC 61850-7-3 et l'IEC 61850-7-2. Les instances sous-jacentes de GenLogicalNodeClass, GenDataObjectClass et GenDataAttributeClass ont un espace de noms implicite (c'est-à-dire IEC 61850-7-4, IEC 61850-7-3 et IEC 61850-7-2) qui est dérivé des références normatives de l'IEC 61850-7-4.

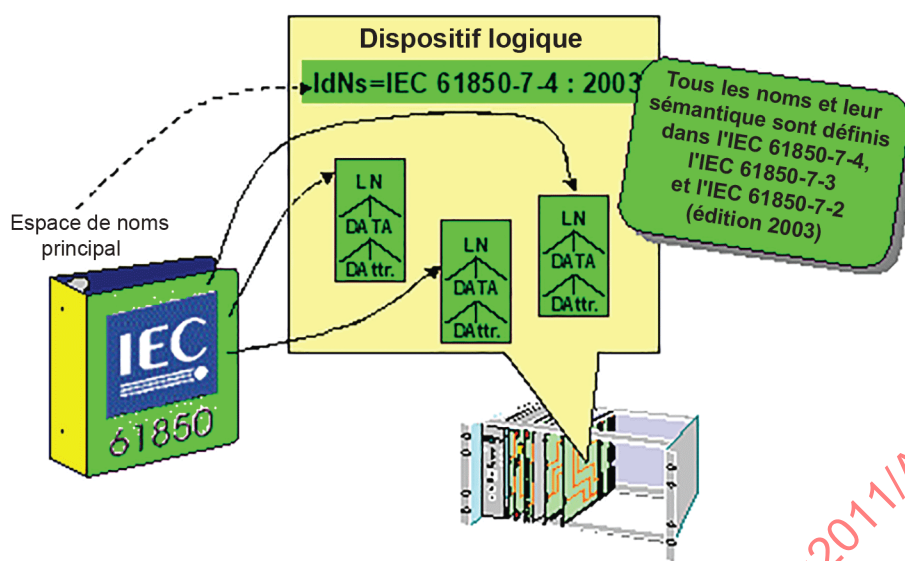


Figure 74 – Toutes les instances dérivées de classes dans un même espace de noms

Lorsque des extensions (facultatives ou obligatoires) sont définies par la version suivante d'une norme, l'espace de noms IdNs de la norme est conservé et une identification de version/révision est ajoutée afin qu'un utilisateur humain puisse trouver la signification. Cela est nécessaire pour la rétrocompatibilité entre les applications. Cela signifie toutefois que toutes les applications appliquent le principe "may ignore" (peut ignorer). Autrement dit, elles ignorent ce qu'elles ne comprennent pas, et il convient que cela soit normalement le cas, car une application automatique utilise juste ce qui lui est nécessaire et ce qu'elle connaît. Des problèmes ne peuvent survenir qu'avec des applications "génériques" comme les pilotes de communication purs qui identifient soudainement de nouvelles CDC ou FC et doivent alors les "ignorer". Des mesures spécifiques doivent être prises pour résoudre les problèmes de compatibilité entre les différentes versions d'une norme. Pour l'IEC 61850, ces exigences peuvent être consultées dans l'Annexe K normative. Pour les extensions "privées" et les domaines d'application supplémentaires, un espace de noms spécifique doit toujours être défini.

Les exemples d'instances représentés à la Figure 75 sont issus de trois espaces de noms différents: l'IEC 61850-7-4, l'autre document normatif, et un espace de noms privé (Vestas). Sachant que la majorité des instances est dérivée de l'IEC 61850-7-4, l'espace de noms de dispositif logique IdNs reste encore l'IEC 61850-7-4 (espace de noms principal).

Le nœud logique du bas a été dérivé de l'autre spécification. Il est nécessaire d'indiquer cette information au moyen d'un espace de noms de nœud logique explicite InNs avec la valeur, par exemple, "Autre spécification: année de publication". Les instances sous ce nœud logique sont définies dans l'espace de noms. Les instances de données ont implicitement le même espace de noms. Le troisième espace de noms appliqué est un espace de noms privé: InNs avec la valeur "Vestas".

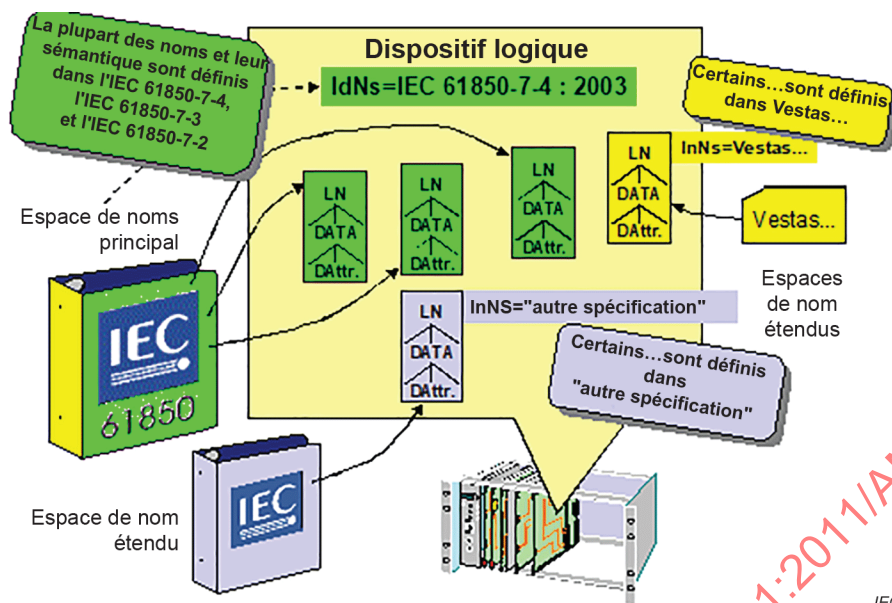


Figure 75 – Instances dérivées de plusieurs espaces de noms

L'espace de noms des dispositifs logiques peut être "IEC 61850-7-4:2003" ou tout autre espace de noms de domaine (tel que défini ci-dessous) dans lequel un dispositif logique est défini et utilisé en fonction du contexte et qui "inclut" au minimum une version de l'IEC 61850-7-4. L'exemple donné à la Figure 76 représente la manière dont une autre spécification peut, par exemple, hériter de toutes les classes (GenLogicalNodeClass, GenDataObjectClass et GenCommonDataClass) de l'IEC 61850-7-4, l'IEC 61850-7-3 et l'IEC 61850-7-2. Dans ce cas, l'autre spécification définirait un espace de noms de noms de surensemble.

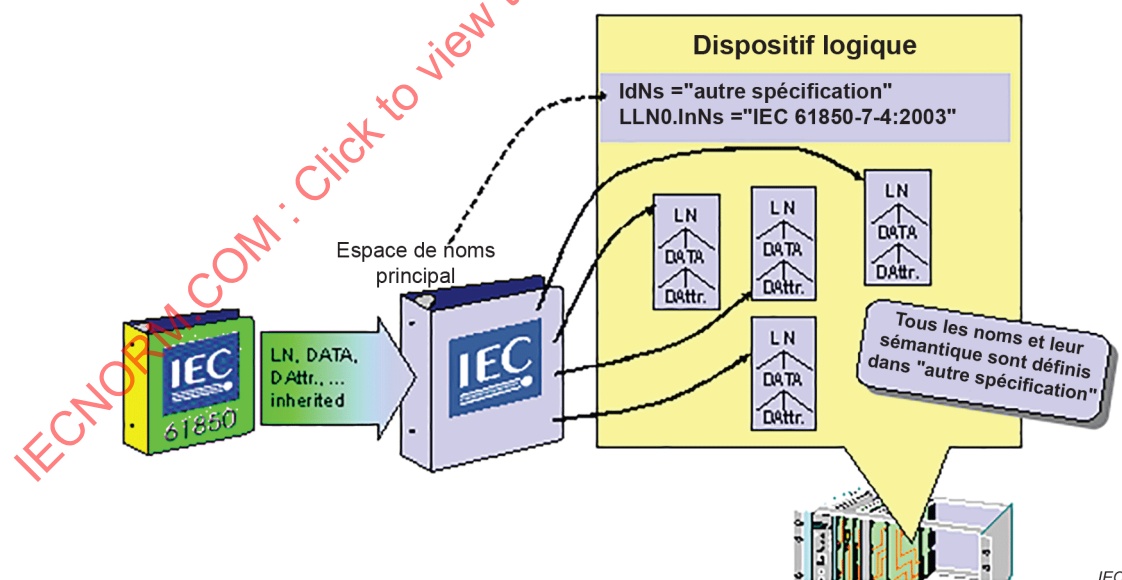


Figure 76 – Espaces de noms hérités

IdNs a la valeur "Autre spécification: année de publication". LLN0.InNs a la valeur "IEC 61850-7-4:2003" qui indique la version des espaces de noms de base sur laquelle est basé le dispositif logique. Puisque toutes les classes sont contenues dans ces espaces de noms, les instances sous-jacentes ont implicitement le même espace de noms. Il n'est pas nécessaire que leurs espaces de noms aient une valeur explicite. Le 13.3 et le 13.4 qui

suivent décrivent en détail les relations entre les espaces de noms détenus par des entités différentes.

La présente norme définit cinq types d'espaces de noms:

- Les **espaces de noms de base** contiennent des définitions des classes de base à partir desquelles tous les autres espaces de noms sont construits. Il s'agit de l'IEC 61850-7-2, de l'IEC 61850-7-3 et de l'IEC 61850-7-4.
- Les **espaces de noms de domaine** incluent les Parties 7-4xx de la série IEC 61850, et éventuellement d'autres espaces de noms hébergés dans des normes dédiées ou des spécifications techniques telles que l'IEC 61850-80-4 et l'IEC 61400-25,
- Les **espaces de noms de norme de produit** contiennent généralement le modèle d'information complet de la plaque signalétique, mais peuvent également s'appliquer à d'autres produits plus génériques. L'IEC 62771-3 Ed 2 pour l'appareillage de commutation ou l'IEC 61869-9 pour les unités de fusion sont des exemples types d'espaces de noms de produit.
- Les **espaces de noms transitoires** sont le plus souvent utilisés dans les rapports techniques tels que les Parties 90-xx de la série IEC 61850. Ces rapports techniques peuvent étendre n'importe quel espace de noms de base (IEC 61850-7-2, IEC 61850-7-3 et IEC 61850-7-4) et n'importe quel espace de noms de domaine. Ils définissent les extensions qui sont supposées être incluses ultérieurement dans le cadre des espaces de noms de base ou de domaine de l'IEC 61850.
- Les **espaces de noms privés** sont développés pour fournir une extension au modèle de données privé (c'est-à-dire un ensemble d'objets de données non normalisés ajouté à l'ensemble normalisé). Leur construction peut inclure des classes de données communes normalisées issues des espaces de noms normalisés ou des extensions d'espace de noms normalisés. Ils sont supposés être gérés par l'entité privée qui les a produits.

13.3 Dépendances d'espaces de noms

13.3.1 Généralités

Trois types de dépendances sont définis:

- **Depends on:** indique qu'une version d'espace de noms ne peut pas exister sans les classes définies dans la version d'espace de noms dont elle dépend. Ce type de dépendance est principalement utilisé pour les espaces de noms de base.
- **Includes:** indique qu'une version d'espace de noms inclut certaines ou l'ensemble des classes définies dans une autre version d'espace de noms. Ce type de dépendance est applicable aux espaces de noms de domaine.
- **Extends:** indique qu'une version d'espace de noms peut étendre les classes définies dans une autre version d'espace de noms. Applicable aux espaces de noms de norme de produit et transitoires.

NOTE "Depends on", "Includes" et "Extends" sont des références de code machine et ne doivent pas être traduites.

13.3.2 Dépendances d'espaces de noms de base

NOTE L'Annexe I informative est une suggestion pour la troisième édition de cette partie de la série IEC 61850 visant à modifier les dépendances entre les espaces de noms relatifs à l'IEC 61850 afin de permettre aux espaces de noms d'application de domaine d'évoluer indépendamment les uns des autres. Elle remplacerait le présent paragraphe.

Dans le cadre de l'IEC, les espaces de noms sont définis par les entités (par exemple le CE 57 de l'IEC) chargées des vocabulaires utilisés par des applications spécifiques au domaine. Ces entités sont appelées propriétaires d'un espace de noms. Les entités peuvent détenir plusieurs espaces de noms.

La Figure 77 représente les dépendances entre les différents espaces de noms dans le cadre de la série IEC 61850.

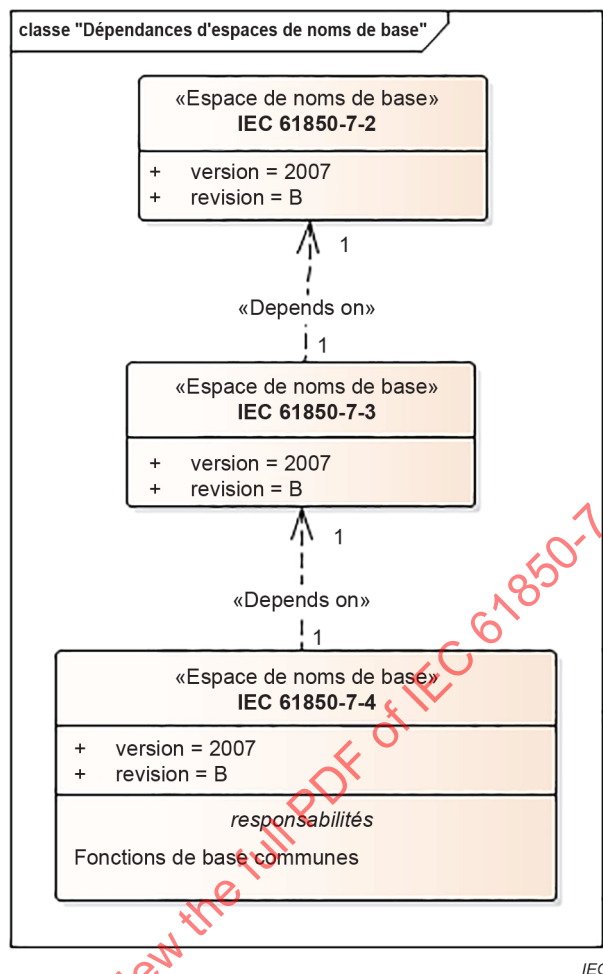


Figure 77 – Dépendances d'espaces de noms de base

Les espaces de noms de base couverts par la série IEC 61850 sont les suivants:

- IEC 61850-7-2, qui spécifie:
 - les types de base, les types de domaines, les types de services et les énumérations;
 - les classes génériques et leurs services associés (classe de nœud logique, classe d'objets de données, classe de données communes);
 - le jeu de données, le bloc de contrôle et les classes de modèle de contrôle;
 - le modèle de classe Événement générique de poste (GSE);
 - le modèle de classe Transmission de valeurs échantillonnées;
 - le modèle Temps et synchronisation du temps;
- IEC 61850-7-3, qui spécifie les classes de données communes et les énumérations;
- IEC 61850-7-4, qui spécifie le LN et les énumérations communes à l'ensemble des applications (et qui héberge actuellement le domaine d'application du poste).

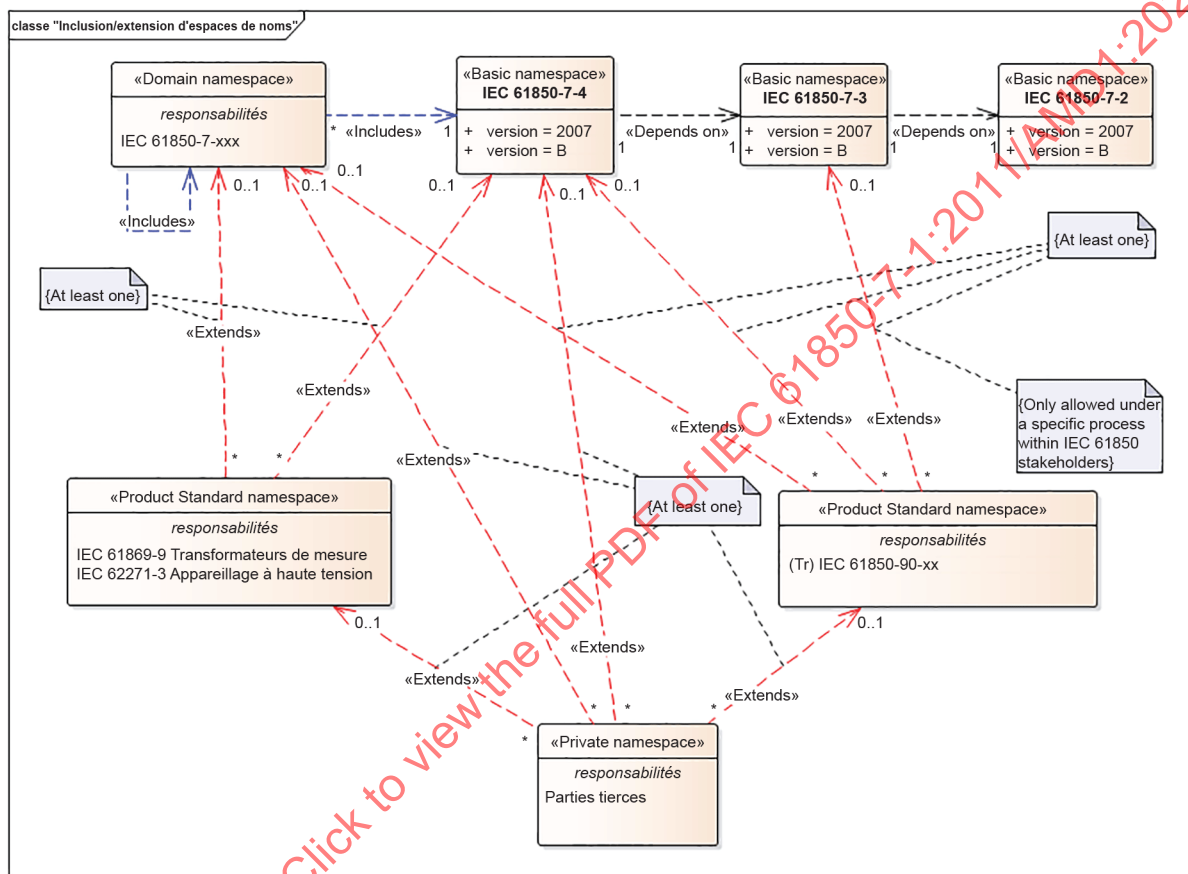
La dépendance "Depends on" de la Figure 77 indique que l'IEC 61850-7-4:2007B, l'IEC 61850-7-3:2007B et l'IEC 61850-7-2:2007B doivent exister dans leur ensemble.

Les espaces de noms applicables aux applications du domaine couverts par la série IEC 61850 sont les suivants:

- IEC 61850-7-410, qui spécifie le LN et les énumérations pour les applications de centrales hydroélectriques;
- IEC 61850-7-420, qui spécifie le LN et les énumérations pour les applications de ressources d'énergie distribuées;
- IEC 61850-7-XXX, pour des applications futures.

13.3.3 Autres dépendances d'espaces de noms

La Figure 78 représente le concept d'inclusion/extension des espaces de noms.



IEC

Figure 78 – Autres dépendances d'espaces de noms

Une application de domaine réutilise de facto certaines ou l'ensemble des classes définies dans l'espace de noms de base IEC 61850-7-4. Cette capacité est représentée par la dépendance "Includes". La spécification d'application de domaine doit indiquer de manière explicite quelle édition/version et révision de l'IEC 61850-7-4 elle inclut. Par exemple, si tel est le cas, l'IEC 61850-420 (Ressources d'énergie distribuées) doit déclarer qu'elle inclut les définitions de la révision B de l'IEC 61850-7-4 version 2007. Un exemple peut être consulté en I.9.

Les extensions (représentées par la dépendance "Extends") aux définitions de la série IEC 61850 peuvent être réalisées par différents acteurs, par exemple:

- Extensions privées réalisées par des tierces parties.
- Extensions réalisées par l'intermédiaire des rapports techniques de l'IEC 61850 et/ou de la nouvelle version/révision des définitions de la série IEC 61850;
- Extensions réalisées par les CE responsables des propriétaires de normes de produit.

Dans la mesure où les espaces de noms évoluent dans le temps et s'agissant de l'amélioration de leur contenu ou des corrections d'erreurs, les propriétaires doivent définir certaines règles permettant d'éviter les problèmes de compatibilité entre les différentes versions d'un espace de noms. Dans le cadre de l'IEC 61850, ces règles peuvent être consultées dans l'Annexe K normative du présent document.

Par ailleurs, des règles d'extensibilité doivent être fournies par les propriétaires afin de permettre à des tierces parties d'étendre un espace de noms sans compromettre l'interopérabilité. Se référer à la spécification IEC TS 61850-1-2 qui fournit des lignes directrices pour l'extension de l'IEC 61850.

L'Article 14 spécifie les règles applicables au contrôle des versions de classes et à l'extension des classes, y compris l'utilisation des espaces de noms.

13.4 Spécification et versionnage des espaces de noms

Les noms d'espace de noms normalisés doivent être structurés comme suit.

NamespaceID:VersionRevision

où

- NamespaceID est l'identification de l'espace de noms. NamespaceID doit se référer à une spécification technique (spécification papier, schéma (XML, base de données), description UML, etc.),
 - Pour les documents normatifs:
 - NamespaceID = IEC {numéro de référence IEC}, par exemple, IEC 61850-7-4
 - Pour les rapports techniques:
 - NamespaceID = (Tr)IEC {numéro de référence IEC}, par exemple, (Tr)IEC 61850-90-4. De plus, le texte d'accompagnement suivant doit être ajouté à la définition du nom d'espace de noms:

"Le contenu de cet espace de noms doit être considéré comme un espace de noms transitoire. Cela signifie que le contenu peut être transféré à un statut de Norme internationale moyennant quelques changements, ce qui rend possibles des rétroincompatibilités avec des mises en œuvre existantes."
- Les espaces de noms transitoires doivent être dépréciés après leur mise en œuvre dans un espace de noms de domaine (voir 13.6). Le document peut être conservé en tant qu'emplacement réservé pour les descriptions de cas d'utilisation,
- La version fait référence à une année arbitraire; il convient qu'elle soit aussi proche que possible de l'année de première publication de l'édition concernée et elle ne doit pas changer en cas d'amendement et de corrigendum de la même édition. Pour le domaine couvert par la première édition de l'IEC 61850-7-4, de l'IEC 61850-7-3 et de l'IEC 61850-7-2, la valeur par défaut est 2003. Pour la deuxième édition, la valeur doit être 2007.
 - "Revision" se réfère à un indicateur de révision. Cet indicateur est exigé pour distinguer les révisions d'une spécification entre deux versions. Pour une première version donnée, la valeur de cet indicateur doit être A et elle est facultative; par exemple, 2007 et 2007A sont jugées équivalentes. Pour le présent amendement de la deuxième édition, la valeur doit être B et elle est obligatoire. Les valeurs de révision suivantes doivent être C, D, ...

Pour les besoins du versionnage, un numéro de version est ajouté à la définition des espaces de noms dans le but de capturer les "micro" évolutions des espaces de noms, en particulier pour prendre en compte les corrections nécessaires dans le cadre des problèmes d'interopérabilité. Le paramètre de version est un nombre incrémentiel entre les révisions (entier commençant à 0 – la mention de la valeur par défaut 0 est facultative). Le numéro de version ne doit pas être utilisé comme partie du nom d'espace de noms normalisé utilisé

comme référence aux espaces de noms (tel que défini en 13.5.1 et lors de l'expression des dépendances d'espaces de noms).

Les noms d'espace de noms doivent être définis de manière explicite dans le support de description du contenu (spécification papier, schéma (XML, base de données), description UML, ...).

13.5 Attributs pour les références à des espaces de noms

13.5.1 Généralités

Les cinq attributs suivants qui référencent les espaces de noms sont définis:

- L'attribut espace de noms de dispositif logique (IdNs) doit référencer l'espace de noms d'application de domaine utilisé comme spécification technique principale pour l'ensemble du dispositif logique;
- L'attribut espace de noms de nœud logique (InNs) du LLN0 doit référencer la version/révision de l'espace de noms de base utilisée pour l'ensemble du dispositif logique.
- L'attribut espace de noms de nœud logique (InNs) doit référencer l'espace de noms de nœud logique d'une seule instance d'une classe de nœuds logiques compatible.
- L'attribut d'espace de noms de données (dataNs) doit référencer l'espace de noms de données d'une seule instance d'une classe d'objets de données compatible.
- l'attribut espace de noms de classe de données communes (cdcNs). Cet attribut de données a été déprécié.

Les classes de données communes contiennent les DataAttributes IdNs et InNs issus de l'extrait du Tableau 14.

**Tableau 14 – Extrait de la classe de données communes
Plaque signalétique de nœud logique (LPL)**

cdcId = LPL, nom de classe UML = LPL					
Nom d'attribut	Type d'attribut	FC	TrgOp	(Valeur/plage de valeurs) Explication	PresCond
DataAttribute pour configuration, description et extension					
...	...	...	...	...	...
IdNs	VisString255	EX		Espace de noms de dispositif logique, par exemple "IEC 61850-7-4:2007B".	MFIn0
InNs	VisString255	EX		Espace de noms de nœud logique.	MOInNs

NOTE 1. Les conditions de présence de la dernière colonne sont définies dans l'IEC 61850-7-2.

Le type abstrait BasePrimitiveCDC regroupe les attributs communs à l'ensemble des classes de données communes. Il contient les DataAttributes cdcName et dataNs, comme indiqué dans l'extrait du Tableau 15.

**Tableau 15 – Extrait du type abstrait
BasePrimitiveCDC de classe de données communes**

Nom de classe UML = BasePrimitiveCDC					
Nom d'attribut	Type d'attribut	FC	TrgOp	(Valeur/plage de valeurs) Explication	PresCond
DataAttribute pour configuration, description et extension					
...	...	...	...	...	...
cdcName	VisString255	EX		Nom de la classe de données communes.	O
dataNs	VisString255	EX		Espace de noms de données.	MOdataNs

NOTE 2 Les conditions de présence de la dernière colonne sont définies dans l'IEC 61850-7-2.

13.5.2 Attribut pour l'espace de noms de dispositif logique (IdNs)

Le DataAttribute espace de noms de dispositif logique IdNs doit être utilisé pour indiquer l'espace de noms de l'application du domaine qui a été utilisé comme espace de noms principal pour l'ensemble du dispositif logique, par exemple IEC 61850-7-4, IEC 61850-7-410, IEC 61850-7-420, etc.

L'attribut IdNs doit être un DataAttribute de la plaque signalétique NamPlt de la classe de nœuds logiques LLN0. L'attribut IdNs doit être défini dans la classe de données communes LPL (plaque signalétique de nœud logique) de l'IEC 61850-7-3. Sa valeur doit être celle spécifiée en 13.4.

L'attribut IdNs doit être disponible dans chaque instance de la classe de nœuds logiques LLN0.

L'ObjectReference pour le DataAttribute IdNs doit être:

LDName/LLN0.NamPlt.IdNs

13.5.3 Attribut pour l'espace de noms de base de dispositif logique (LLN0.NamPlt.InNs)

Le DataAttribute espace de noms de nœud logique LLN0.NamPlt.InNs doit être utilisé pour indiquer l'espace de noms de base utilisé pour un dispositif logique. La version/révision de cet espace de noms de base impose la version/révision de l'espace de noms de base qui est mise en œuvre dans le dispositif. Des exemples peuvent être consultés en J.9 et J.7.

L'attribut doit être disponible dans une instance de LLN0 si l'espace de noms de base utilisé diffère de l'espace de noms du dispositif logique. L'attribut est facultatif au cas où l'espace de noms de base utilisé serait identique à l'espace de noms du dispositif logique.

L'attribut InNs doit être un DataAttribute de la plaque signalétique NamPlt d'un nœud logique LLN0. L'attribut InNs doit être défini dans la classe de données communes LPL (plaque signalétique de nœud logique) de l'IEC 61850-7-3 versionnée/révisée incluse dans l'espace de noms de base.

L'ObjectReference pour le DataAttribute InNs doit être:

LDName/LNN0.NamPlt.InNs

13.5.4 Attribut pour l'espace de noms de nœud logique (lnNs)

Le DataAttribute espace de noms de nœud logique lnNs doit être utilisé pour indiquer l'espace de noms qui spécifie la classe de nœuds logiques. L'attribut doit être disponible dans une instance de LN si sa valeur diffère de la valeur de ldNs. L'attribut est facultatif au cas où la valeur de lnNs serait identique à la valeur de ldNs.

L'attribut lnNs doit être un DataAttribute de la plaque signalétique NamPIt d'un nœud logique. L'attribut lnNs doit être défini dans la classe de données communes LPL (plaque signalétique de nœud logique) de l'IEC 61850-7-3 versionnée/révisée incluse dans l'espace de noms de base.

L'ObjectReference pour le DataAttribute lnNs doit être:

LDName/LNName.NamPIt.lnNs

La valeur de lnNs doit être:

- l'espace de noms normalisé où le nœud logique est défini;
- l'espace de noms normalisé où la classe de nœuds logiques a été étendue dès que l'instance de nœud logique a utilisé l'extension^{1,2};
- l'espace de noms privé si le nœud logique a été défini hors de l'espace de noms normalisé.

13.5.5 Attribut d'espace de noms de données (dataNs)

Le DataAttribute espace de noms de données dataNs doit être utilisé pour indiquer l'espace de noms qui spécifie l'utilisation d'une classe d'objets de données compatible. L'attribut est facultatif pour les classes d'objets de données de la classe DomainLN (NamPIt, Health, Mod, par exemple)³. Pour toutes les autres classes d'objets de données, l'attribut doit être disponible dans une instance de LN si sa valeur diffère de la valeur de lnNs. L'attribut est facultatif au cas où la valeur de dataNs serait identique à la valeur de lnNs.

L'attribut dataNs doit être un DataAttribute de la donnée. L'attribut dataNs doit être défini dans les classes de données communes de l'IEC 61850-7-3.

L'ObjectReference pour le DataAttribute dataNs doit être:

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

13.5.6 Attribut pour l'espace de noms de classe de données communes (cdcNs)

Cet attribut a été déprécié.

13.5.7 Attribut pour la dénomination d'un nom de classe de données communes (cdcName)

Le DataAttribute nom de la classe de données communes cdcName peut être utilisé pour identifier une CDC dans le cas de l'extension du modèle, par exemple dans un rapport technique. Un exemple peut être consulté en J.7.

¹ La modification d'une condition de présence dans une norme de produit est considérée comme une extension.

² L'extension normative d'une énumération est considérée comme une extension.

³ DomainLN est défini dans l'IEC 61850-7-4.

13.6 Dépréciation des espaces de noms

Les espaces de noms ont un cycle de vie, et la dépréciation des espaces de noms peut être motivée par deux raisons principales:

- Le contenu de l'espace de noms considéré a été déplacé dans d'autres parties, généralement un espace de noms transitoire dont le contenu a migré vers une nouvelle édition/version des espaces de noms de domaine ou de base, devient de facto déprécié.
- Une nouvelle édition/version d'un espace de noms donné est disponible et rend de facto le précédent déprécié

La dépréciation d'un espace de noms donné signifie qu'il n'est plus recommandé de construire de nouveaux espaces de noms sur celui déprécié.

Ces espaces de noms dépréciés sont toujours disponibles pour l'interopérabilité. Toutefois, il n'y a plus de gestion, c'est-à-dire qu'il n'est plus possible de soulever des "problèmes techniques" sur ces espaces de noms. Les éventuels problèmes techniques concernant un espace de noms déprécié peuvent déjà avoir été corrigés dans une nouvelle édition/version. Dans le cas contraire, il importe de se référer à la dernière édition/version.

14 Règles communes pour une nouvelle version de classes et pour l'extension des classes d'objets

Remplacer le texte, les figures et les tableaux existants de l'Article 14 par les nouveaux textes, figures et tableaux suivants:

14.1 Généralités

Le présent article spécifie les règles pour:

- le contrôle des versions de classes,
- l'extension des classes,
- l'utilisation de versions de classes ultérieures (utilisation de la postdéfinition).

La description des règles est prise en charge par l'utilisation d'exemples de scénarios de cas d'utilisation en Annexe J.

Les règles s'appliquent aux classes suivantes: classes de nœuds logiques (LN), classes de données, classes de données communes (CDC), énumérations et classes de blocs de contrôle.

Les règles sont présentées dans l'ordre suivant:

- règles de base;
- extensions réalisées dans l'espace de noms privé;
- extensions réalisées dans les espaces de noms de norme de produit;
- extensions réalisées dans les espaces de noms transitoires;
- extensions réalisées dans les espaces de noms de base et de domaine.

Les classes faisant partie d'un espace de noms de propriétaire (par exemple, IEC 61850-7-4) sont appelées "classes normalisées" dans les articles et paragraphes suivants.

14.2 Règles de base

14.2.1 Généralités

14.2.1.1 Règles de base

Les règles de base sont décrites dans le diagramme de la Figure 78.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

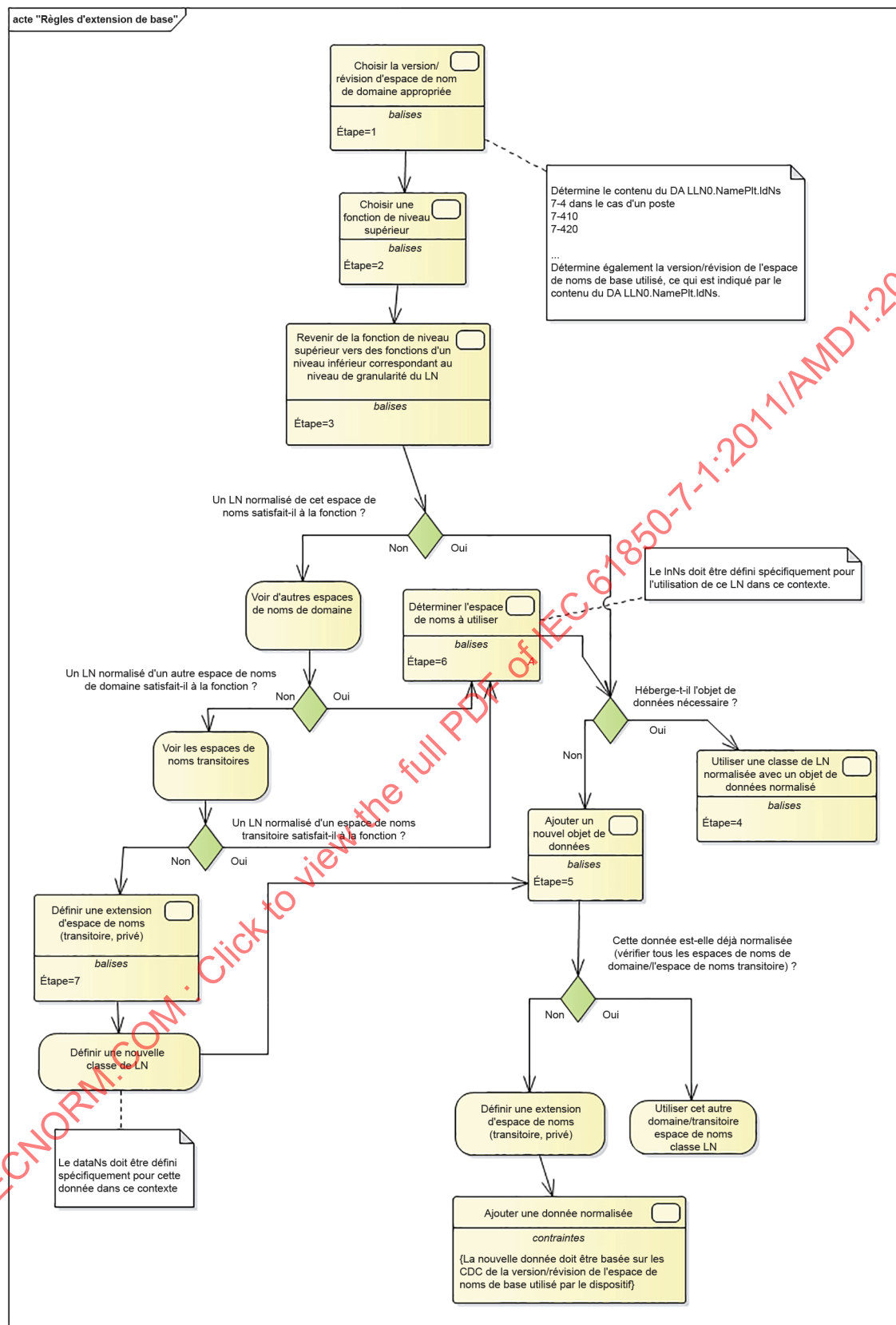


Figure 78 – Diagramme des règles d'extension de base

La première étape consiste à choisir la version/révision d'un espace de noms de domaine cible. Cette étape permet de déterminer l'espace de noms du dispositif logique identifié par l'attribut de données LLN0.NamPlt.IdNs. Elle impose également la version/révision des espaces de noms de base supposés être mis en œuvre dans le dispositif. Cette version/révision des espaces de noms de base doit être spécifiée par l'attribut NamPlt.InNs du LN LLN0. Voir 13.5.3.

La deuxième étape consiste à choisir une fonction de niveau supérieur à modéliser. La troisième étape consiste à décomposer la fonction d'application exigée à un degré de granularité des classes de nœuds logiques existantes du domaine d'application (par exemple, l'automatisation de poste dans le cas de la série IEC 61850) qui sont répertoriées dans l'IEC 61850-7-4. Les règles décrites dans les paragraphes suivants doivent alors être appliquées.

14.2.1.2 Utilisation de classes de LN normalisées avec des classes d'objets de données normalisées (étape 4)

Lorsqu'une classe de LN normalisée figurant dans l'espace de noms de domaine choisi remplit la fonction, elle doit être utilisée et les règles suivantes doivent s'appliquer:

- a) S'il existe une classe de nœuds logiques qui s'ajuste à la fonction à modéliser, une instance de ce nœud logique doit être utilisée avec toutes ses données obligatoires (indiquées par les conditions de présence dans le tableau de LN). Les règles d'une instantiation unique peuvent être consultées dans l'IEC 61850-7-2.
- b) Si, en plus des données obligatoires, il existe aussi des données facultatives (indiquées par les conditions de présence dans le tableau de LN), qui correspondent à la fonction à modéliser, ces données facultatives doivent être utilisées.
- c) Les instances de données doivent être utilisées avec tous leurs attributs de données obligatoires (indiqués par les conditions de présence dans le tableau de CDC). S'il existe aussi des attributs de données facultatifs (indiqués par les conditions de présence dans le tableau de CDC), qui correspondent à la fonction à modéliser, ces attributs de données facultatifs doivent être utilisés.
- d) S'il existe les mêmes données qu'il est nécessaire d'instancier plusieurs fois, des données complémentaires avec des extensions numériques doivent être utilisées conformément aux règles applicables à la spécialisation de données en utilisant les extensions numériques (voir 14.8).
- e) S'il existe des versions dédiées de la fonction à modéliser avec les mêmes données de base (par exemple terre, phase, zone A, zone B, etc.), des instances différentes de cette classe de nœuds logiques doivent être utilisées (voir 14.7).
- f) Les définitions de classes de LN ultérieures (postdéfinitions⁴) peuvent être utilisées à condition que la version/révision des CDC soit identique à celle de l'espace de noms de base utilisé (voir J.5).

14.2.1.3 Extension de classes de LN normalisées avec de nouvelles classes d'objets de données (étape 5)

Lors de l'extension de classes de LN normalisées avec de nouvelles classes d'objets de données, les règles suivantes doivent s'appliquer:

- a) Si une classe d'objets de données avec la même sémantique, dans le contexte de la classe de LN, est déjà normalisée dans un autre espace de noms de domaine/transitoire, cette autre classe de LN de domaine/espace de noms transitoire doit être utilisée (Voir J.4).

⁴ La postdéfinition d'une classe, est l'utilisation d'une classe qui a été définie dans une version/révision ultérieure de l'espace de nom du domaine d'application spécifié.

- b) Sinon, une nouvelle classe d'objets de données basée sur les CDC de la version/révision de l'espace de noms de base utilisé par le dispositif doit être créée dans un espace de noms transitoire/privé.

14.2.1.4 Utilisation de classes de LN normalisées issues d'autres espaces de noms (étape 6)

Lorsqu'aucune classe de LN normalisée de l'espace de noms de domaine choisi ne remplit la fonction et qu'une classe de LN issue d'un autre espace de noms de domaine/transitoire remplit cette fonction, elle doit être utilisée.

14.2.1.5 Extension d'un espace de noms de domaine avec de nouvelles classes de LN (étape 7)

Lors de l'extension d'un espace de noms de domaine avec de nouvelles classes de LN, les nouvelles classes de LN doivent être créées dans un espace de noms transitoire/privé.

Les nouvelles classes de LN héritent de toutes les classes d'objets de données de la classe DomainLN. Pour ces objets de données, la présence de l'attribut dataNs est facultative. S'il est présent, il doit avoir la même valeur que le LLN0.NamPit.InNs⁵ applicable.

14.2.2 Utilisation d'informations facultatives

La condition de présence de plusieurs classes d'objets de l'IEC 61850 est définie comme étant facultative. Par exemple, certains LN tels que MMXU n'ont même pas d'objets de données spécifiques. Ceci est en partie attribuable à l'accord global en ce qui concerne la modélisation de la granularité. Dans l'exemple de MMXU, plutôt que d'avoir des dizaines de LN relatifs aux mesures fonctionnelles ou statistiques, il n'existe qu'un seul LN comportant des dizaines d'objets de données facultatifs qui sont utilisés selon la mesure exigée. Ce qui soulève la question du mode d'utilisation des informations facultatives.

Que ce soit dans le contexte d'un utilisateur final ou d'un fabricant de systèmes/dispositifs, les objets de données ou les attributs de données doivent être utilisés dans le but de remplir une fonction bien définie. Il convient d'utiliser uniquement les objets exigés. Dans le cas des profils d'application de base (BAP définis dans l'IEC 61850-7-6) qui décrivent les profils dédiés à un groupe d'utilisateurs, l'ensemble des objets mentionnés comme étant obligatoires est pris en compte comme tel, même si la norme affirme le contraire.

Il convient que les dispositifs (serveurs et clients) soient suffisamment souples concernant l'utilisation d'informations facultatives dans les rapports. Les serveurs doivent permettre de configurer les jeux de données de manière flexible au niveau des données (FCD). Les clients doivent être suffisamment souples pour recevoir des attributs (FCDA) dont les fonctionnalités associées ne sont pas prises en charge par le client. L'IEC 61850-7-2 peut être consultée pour obtenir une explication sur la production de rapports et l'inclusion des membres de jeu de données.

14.2.3 Règles générales pour les types d'énumérations et les attributs et objets de données basés sur l'énumération

Les règles définies dans l'IEC 61850-7-2 doivent être suivies.

⁵ Voir l'IEC 61850-7-4 pour une définition de la classe DomainLN.

14.3 Extensions réalisées dans un espace de noms privé

14.3.1 Règles pour les classes LN

Les extensions aux classes LN normalisées peuvent être réalisées par des tierces parties. Cependant, les tierces parties ne peuvent pas modifier la sémantique d'un LN et par conséquent ne peuvent pas modifier le nom de l'espace de noms lors de la réalisation des extensions. Afin de ne pas compromettre l'interopérabilité, les règles relatives aux extensions LN sont les suivantes.

- a) Le nouveau nom de classe d'objets de données doit respecter les conventions de dénomination définies dans l'IEC 61850-7-2. En outre, l'attribut dataNs doit être obligatoire et être un espace de noms privé.
- b) La nouvelle classe d'objets de données doit être allouée à une et une seule des classes de données communes définies dans l'IEC 61850-7-3.
- c) Les classes d'objets de données normalisées peuvent être utilisées. Dans ce cas (voir Article J.3):
 - La sémantique doit être celle de la classe d'objets de données normalisée.
 - dataNs doit être obligatoire et être un espace de noms privé.
 - La CDC doit être identique à celle de la classe d'objets de données normalisée.
- d) Les définitions de classe d'objets de données normalisées ultérieures (postdéfinitions) peuvent être utilisées. Dans ce cas (voir Article J.4):
 - La sémantique et le type doivent être ceux de la classe d'objets de données normalisée.
 - Dans le cas d'une extension de classe LLNO, l'attribut dataNs doit être obligatoire et identique à celui de la classe d'objets de données normalisée.
 - Pour d'autres classes de LN compatibles, la définition suivante de la classe de LN doit être utilisée (nouvelle version/révision de l'InNs)
- e) La description de la nouvelle classe d'objets de données doit être ajoutée à la documentation IEC du système spécifique fourni ou au projet spécifique au client.

La définition de classes d'objets de données privées peut avoir des inconvénients et il convient de prendre connaissance de la remarque suivante.

Dès qu'un espace de noms normalisé introduit une classe d'objets de données privée, sitôt que le dispositif est déclaré comme étant conforme à la nouvelle version de l'espace de noms, alors:

- Si la CDC utilisée et la sémantique de l'objet de données correspondent à la norme, il importe que le dispositif soit conforme à la nouvelle classe d'objets de données normalisée et l'attribut dataNs doit être l'attribut normalisé.
- Si la sémantique de la CDC ne correspond pas, l'attribut dataNs doit alors rester privé et la classe d'objets de données privée doit être renommée.

14.3.2 Nouvelles classes de LN

Des tierces parties peuvent étendre le jeu de classes de LN du propriétaire en définissant de nouveaux LN hors de l'espace de noms du propriétaire. Dans ce cas, les règles sont les suivantes:

- a) l'InNs doit être obligatoire et être un espace de noms privé.
- b) La nouvelle classe de LN doit contenir la classe d'objets de données Beh qui doit suivre la définition de la hiérarchie LD.
- c) Les classes d'objets de données normalisées peuvent être réutilisées (voir Figure J.1). Dans ce cas:
 - La sémantique doit être celle de la classe d'objets de données normalisée.

- La CDC doit être identique à celle de la classe d'objets de données normalisée.
- La présence de l'attribut dataNs est facultative. S'il est présent, il doit avoir pour valeur un espace de noms privé.
- La description de la nouvelle classe LN doit être ajoutée à la documentation IEC du système spécifique fourni ou au projet spécifique au client.

La définition de classes de LN privées peut avoir des inconvénients et il convient de prendre connaissance des remarques suivantes.

Dès qu'un espace de noms normalisé introduit la fonctionnalité de la classe de LN privée, sitôt que le dispositif est déclaré comme étant conforme à la nouvelle version de l'espace de noms, alors:

- a) Si la fonctionnalité de la classe de LN privée correspond à la norme, le dispositif doit être conforme à la nouvelle classe de LN normalisée et l'attribut dataNs doit être l'attribut normalisé (les instances de LN privées précédentes peuvent être conservées pour éviter les problèmes de compatibilité descendante au sein du système);
- b) Si la fonctionnalité de la classe de LN privée ne correspond pas, l'attribut lnNs doit alors rester privé et la classe de LN privée doit être renommée.

14.3.3 Règles pour les classes de données communes

A partir de la deuxième édition (2011) de la présente norme, les nouvelles versions des classes de données communes normalisées ne doivent être établies que par le propriétaire de l'espace de noms. L'extension privée d'une CDC existante doit donc être proscrite.

14.3.4 Règles pour les types d'énumérations et les attributs et objets de données basés sur l'énumération

14.3.4.1 Extensions privées pour l'attribut de données normalisé basé sur l'énumération

Les extensions privées qui respectent les règles générales définies en 14.2.3 sont admises, mais peuvent cependant entraîner des problèmes d'interopérabilité.

14.3.4.2 Extensions privées pour les objets de données normalisés basés sur l'énumération

Les extensions privées qui respectent les règles générales définies en 14.2.3 sont admises, mais peuvent cependant entraîner des problèmes d'interopérabilité.

14.3.4.3 Extensions privées pour les objets de données privés basés sur l'énumération

Les objets de données privés peuvent comporter des énumérations privées.

14.4 Extensions réalisées dans les espaces de noms de norme de produit

La série IEC 61850 est un ensemble de normes horizontales qui couvre plusieurs domaines d'application. De ce point de vue, les normes de famille de produits peuvent être considérées comme des normes verticales qui mettent en œuvre certaines parties de la série IEC 61850.

Les CE chargés des normes de famille de produits peuvent réaliser des extensions spécifiques à d'autres espaces de noms (voir Figure 79). L'IEC 61850-1-2 peut être consultée pour obtenir des détails sur les règles applicables aux extensions dans les espaces de noms de norme de produit. Un exemple est donné en J.8.

14.5 Extensions réalisées dans les espaces de noms transitoires

Les espaces de noms transitoires (la série IEC TR 61850-90-XXX) sont destinés à établir des propositions pour les extensions d'autres espaces de noms (voir Figure 79). Ces extensions peuvent ensuite être intégrées dans les parties concernées de la série IEC 61850. Un exemple est donné en J.6.

Les espaces de noms transitoires sont autorisés dans le cadre d'un processus spécifique visant à créer des CDC. Un exemple est donné en J.7. Les espaces de noms transitoires ne peuvent pas étendre les CDC existants. Se référer à l'IEC 61850-1-2 qui fournit des lignes directrices pour les extensions au sein des espaces de noms transitoires.

NOTE Ces extensions ne sont pas soumises aux essais de conformité.

14.6 Extensions réalisées dans les espaces de noms de base et de domaines

Les espaces de noms de domaine contiennent des définitions applicables à certains domaines, tels que l'automatisation de poste ou les ressources d'énergie distribuées. Toute définition contenue dans un espace de noms de domaine peut être réutilisée dans un autre espace de noms de domaine. Avant de réaliser une extension, il convient donc qu'un propriétaire d'espace de noms de domaine vérifie que l'extension n'existe pas déjà dans un autre espace de noms de domaine. L'IEC 61850-1-2 peut être consultée pour obtenir des détails sur les règles applicables aux extensions dans les espaces de noms de base et de domaine.

NOTE Ces extensions sont soumises aux essais de conformité.

14.7 Multiples instances des classes de LN pour des fonctions dédiées et complexes⁶

14.7.1 Exemple pour la protection temporisée de surintensité

Nom de classe de nœuds logiques: PTOC (protection temporisée de surintensité)		
Nom d'instance LN (avec préfixe LN)	Signification	Signification de la valeur de départ StrVal
GndPTOC1	Détection de défaut de terre	"Ground Start Value" (Valeur de départ terre)
PhsPTOC1	Détection de défaut de phase	"Phase Start Value" (Valeur de départ de phase)

La sémantique des différentes instances peut être donnée dans l'attribut description "d" de la donnée NamPlt (plaque signalétique).

14.7.2 Exemple pour la protection PDIS

Nom de classe de nœuds logiques: PDIS (distance)	
Nom d'instance LN (sans préfixe LN)	Signification
PDIS1	Zone 1 de la protection de distance
PDIS2	Zone 2 de la protection de distance
PDIS3	Zone 3 de la protection de distance
Etc.	Etc.

La sémantique des différentes instances peut être donnée dans l'attribut description "d" de la donnée NamPlt (plaque signalétique).

⁶ Les exemples présentés dans cette article ont une valeur informative et ne sont donc pas normatifs.

14.7.3 Exemple pour transformateur de puissance

Nom de classe de nœuds logiques: YPTR (transformateur de puissance)	
Nom d'instance LN (sans préfixe LN)	Signification
YPTR1	Phase d'unité de transformateur L1
YPTR2	Phase d'unité de transformateur L2
YPTR3	Phase d'unité de transformateur L3

La sémantique des différentes instances peut être donnée dans l'attribut description "d" de la donnée NamPlt (plaque signalétique).

14.7.4 Exemple pour réseau auxiliaire

Nom de classe de nœuds logiques: ZAXN (réseau auxiliaire)	
Nom d'instance LN (sans préfixe LN)	Signification
ZAXN1	220 V CC
ZAXN2	60 V CC
ZAXN3	380 V CC

La sémantique des différentes instances peut être donnée dans l'attribut description "d" de la donnée NamPlt (plaque signalétique).

14.8 Spécialisation de données par utilisation d'extensions numériques

Les noms de données normalisés dans les nœuds logiques assurent une identification unique. Si le même objet de données (c'est-à-dire des données ayant la même sémantique) est nécessaire plusieurs fois, comme défini, des objets de données complémentaires avec extensions numériques doivent être utilisés. Les règles relatives aux extensions numériques doivent suivre les conventions de dénomination définies dans l'IEC 61850-7-2 et doivent être comme suit:

- l'utilisation d'extensions numériques ne doit être définie que par le propriétaire de l'espace de noms de données. Pour ce faire, la condition de présence (PresCond) de l'objet de données doit être réglée sur Omulti,
- les objets de données sans condition de présence (PresCond) réglée sur Omulti ne doivent pas faire l'objet d'une extension par des tierces parties,
- les objets de données comportant une condition de présence (PresCond) réglée sur Omulti peuvent faire l'objet d'une extension. Les extensions numériques peuvent être rangées dans un certain ordre ou ne pas l'être (1,2,3,4, ou 1,2,19,25),
- même si une seule instance d'une donnée extensible est présente dans un LN, elle doit avoir une extension numérique.

Des exemples sont donnés ci-après:

Nom de classe de nœuds logiques: PSCH (plan de protection)	
Nom de donnée: RxTr1 (signal de déclenchement direct de téléprotection)	
RxTr1	signal de déclenchement direct de téléprotection 1
RxTr2	signal de déclenchement direct de téléprotection 2
RxTr3	signal de déclenchement direct de téléprotection 3

14.9 Exemples pour de nouveaux LN

Nouveau LN "Automatic door entrance control" (commande automatique d'entrée de porte)

1 ^{er} caractère Indication de groupe de nœuds logiques	2 ^e caractère	3 ^e caractère	4 ^e caractère	Nouveau LN	Valeur de ADEC.NamPIt.InNs
A pour "Automatic Control" (commande automatique)	Door (porte)	Entrance (entrée)	Control (commande)	ADEC "Automatic door entrance control" (commande automatique d'entrée de porte)	myVendor1Extension

Nouveau LN "Fire protection" (protection incendie)

1 ^{er} caractère Indication de groupe de nœuds logiques	2 ^e caractère	3 ^e caractère	4 ^e caractère	Nouveau LN	Valeur de ZFPT.NamPIt.InNs
Z pour "Further equipment" (autre équipement)	Fire (feu)	Protection	Transformer (transformate ur)	ZFPT "Fire protection of a power transformer" (protection incendie d'un transformateur de puissance)	myVendor2Extension

14.10 Exemple pour de nouvelles données

Nouvelle donnée "Colour of Transformer Oil" (couleur de l'huile pour transformateur)

Nom de donnée	Type d'attribut (CDC)	Valeur de ColrTOil.dataNs
ColrTOil	INS	myVendor1Extension

Ajouter, après l'Article 14, le nouvel Article 15 suivant:

15 Compatibilité entre les différentes versions de la norme

Différentes versions de la présente norme seront publiées, principalement en raison de son évolution. Toutes ces versions doivent être compatibles de manière ascendante et descendante les unes avec les autres. Si cette compatibilité n'est pas possible pour une raison quelconque, les exigences générales et les recommandations concernant les problèmes de compatibilité sont fournies à l'Annexe K.

Annexe A (informative)

Vue d'ensemble des nœuds logiques et des données

A.1 Classes de nœuds logiques compatibles et classes de données (IEC 61850-7-4)

A.1.3 Classes d'objets de données (IEC 61850-7-4)

Remplacer le texte existant du A.1.3 par le nouveau texte suivant:

Les classes d'objets de données sont définies dans chaque nœud logique de l'IEC 61850-7-4.

Les noms d'objets de données sont composés en utilisant des abréviations normalisées énumérées dans un tableau au début de l'IEC 61850-7-4, par exemple:

Terme	Description
A	Courant
Acs	Accès
Acu	Acoustique
Age	Vieillessement
Alm	Alarme
Amp	Courant non lié à la phase
An	Analogique
Ang	Angle
...	...

Il convient d'utiliser ces abréviations dans la création de nouveaux noms de données.

La classe de données communes à utiliser avec une entité DATA spécifique est définie dans les nœuds logiques.

Tableau A.1 – Extrait des classes de données pour les mesurandes

Supprimer le Tableau A.1.

Annexe B (informative)

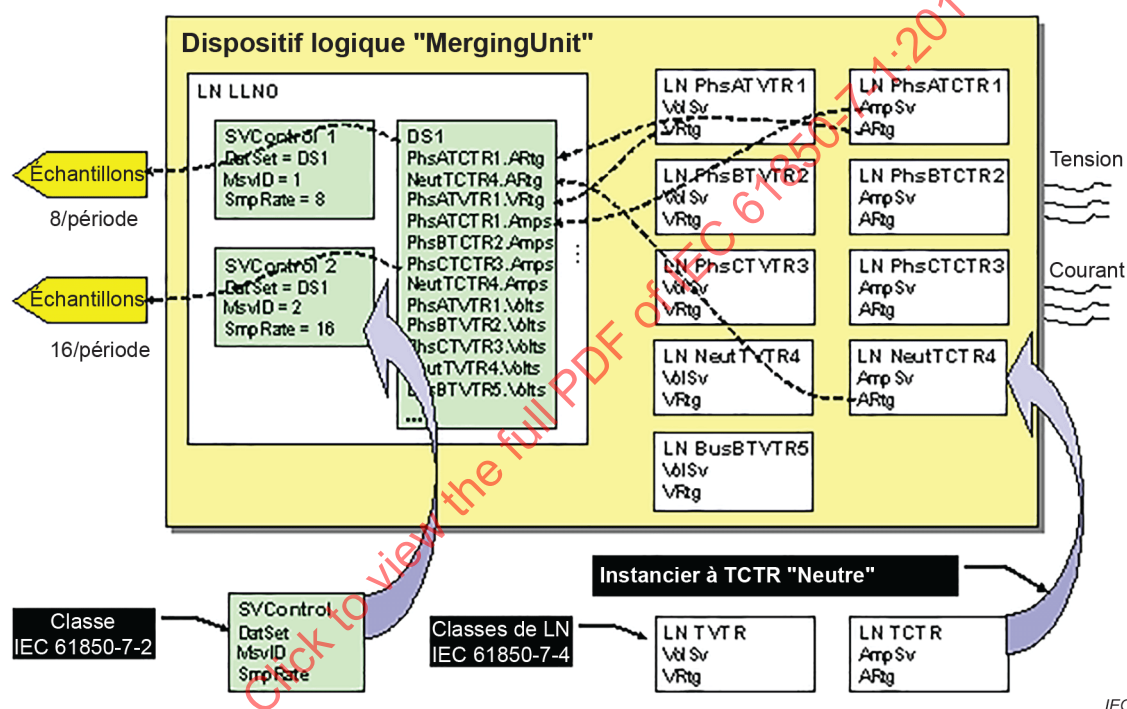
Affectation des données aux nœuds logiques

Remplacer le premier alinéa existant sous la Figure B.1 par le nouveau texte suivant:

Dans le cas où une donnée compatible a été définie dans la norme pour une application spécifique, cette donnée compatible doit être utilisée au lieu de définir une nouvelle donnée, comme spécifié dans les règles d'extension à l'Article 14.

Figure B.3 – Unité de fusion et échange de valeurs échantillonnées (données)

Remplacer la Figure B.3 existante par la nouvelle figure suivante:



Remplacer les deux alinéas existants qui suivent la Figure B.3 par le nouveau texte suivant:

Les valeurs de courant et de tension sont reçues sur le côté droit. Les valeurs triphasées du courant et de la tension sont modélisées dans les nœuds logiques suivants:

- Transformateur de courant – la classe TCTR dans l'IEC 61850-7-4 instanciée pour trois phases et neutre:

PhsATCTR1, PhsBTCTR2, PhsCTCTR3, NeutTCTR4,

- Transformateur de tension – la classe TVTR dans l'IEC 61850-7-4 instanciée pour trois phases, neutre et barre omnibus:

PhsATVTR1, PhsBTVTR2, PhsCTVTR3, NeutTVTR4, BusBTVTR5,

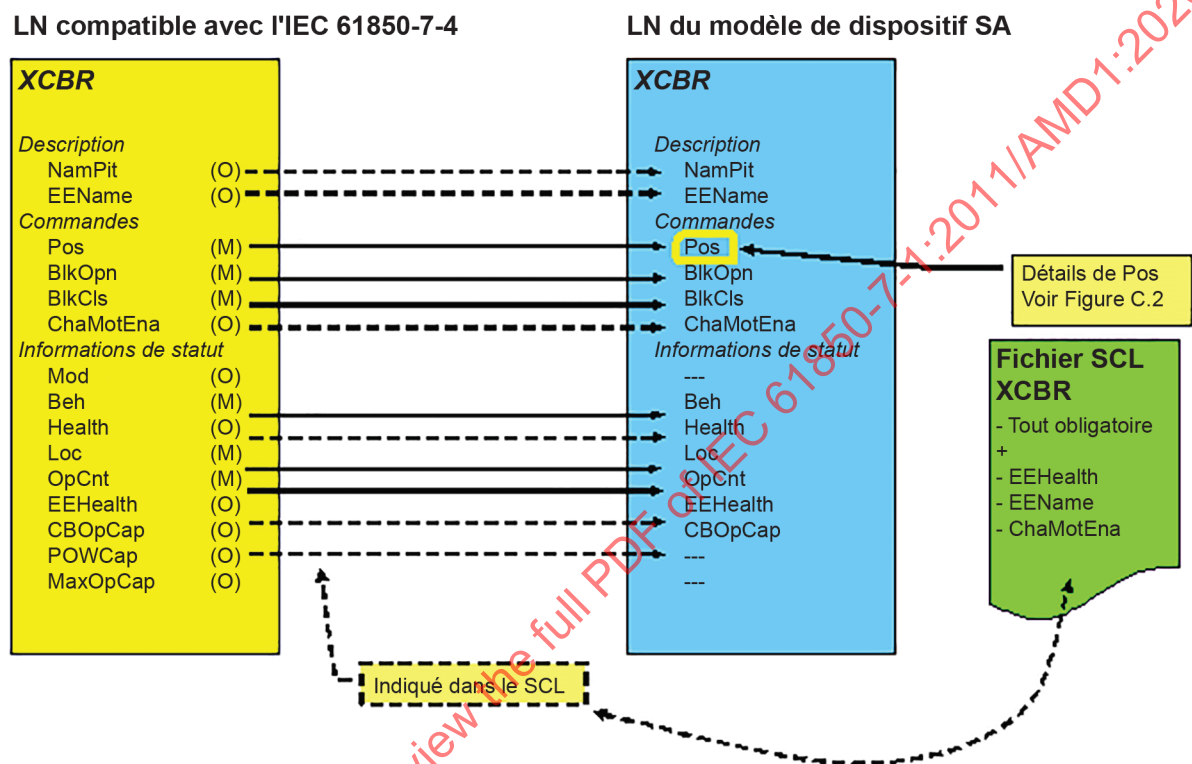
Les valeurs échantillonnées (AmpSv et VolSv) et les caractéristiques nominales correspondantes sont utilisées dans l'exemple. Ces données sont référencées par le jeu de données "DS1".

Annexe C (informative)

Utilisation du langage de configuration de poste (SCL)

Figure C.1 – Application de SCL pour les LN (à titre conceptuel)

Remplacer la Figure C.1 existante par la nouvelle figure suivante:



IEC

Figure C.2 – Application de SCL pour les données (à titre conceptuel)

Remplacer la Figure C.2 existante par la nouvelle figure suivante:

**Classe de données compatible
avec l'IEC 1850-7-4**

Pos (CDC: DPC)		
<i>Statut</i>		
origin	ST	O
ctlNam	ST	O
stVal	ST	M
q	ST	M
t	ST	M
StSeld	ST	MOsbo
<i>Configuration, description et ext.</i>		
pulseConfig	CF	O
ctlModel	CF	O
sboTimeout	CF	M
sboClass	CF	M
d	CF	M
cd:Name	EX	O
dataNs	EX	MOdataNs

**Données du modèle de
dispositif SA**

Pos		
<i>Statut</i>		
origin	= value	
...		
stVal	= value	
q	= invalid/oldData	
t	= value	
...		
<i>Configuration, description et ext.</i>		
ctlModel	= "sbo-with-no-sec."	
sboTimeout	= "5 min"	
sboClass	= "operate, once"	
...		

Détails du fichier SCL pour une instance XCBR		
...		
Tout obligatoire		
+		
-		
...		
ctlModel	= "sbo-with-no-sec."	
sboTimeout	= "5 min"	
sboClass	= "operate, once"	
...		

Indiqué dans le SCL

Valeur
configurée

Figure C.2 – Application de SCL pour les données (à titre conceptuel)

IEC

Annexe D (informative)

Application du concept de LN à des options pour futures extensions

Remplacer le texte existant de l'Annexe D par le nouveau texte suivant:

Cette page est laissée intentionnellement vide.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

Annexe F (informative)

Mapping de l'ACSI à des systèmes de communication réels

Figure F.9 – Message de rapport relatif aux informations de la MMS

Remplacer la Figure F.9 existante par la nouvelle figure suivante:

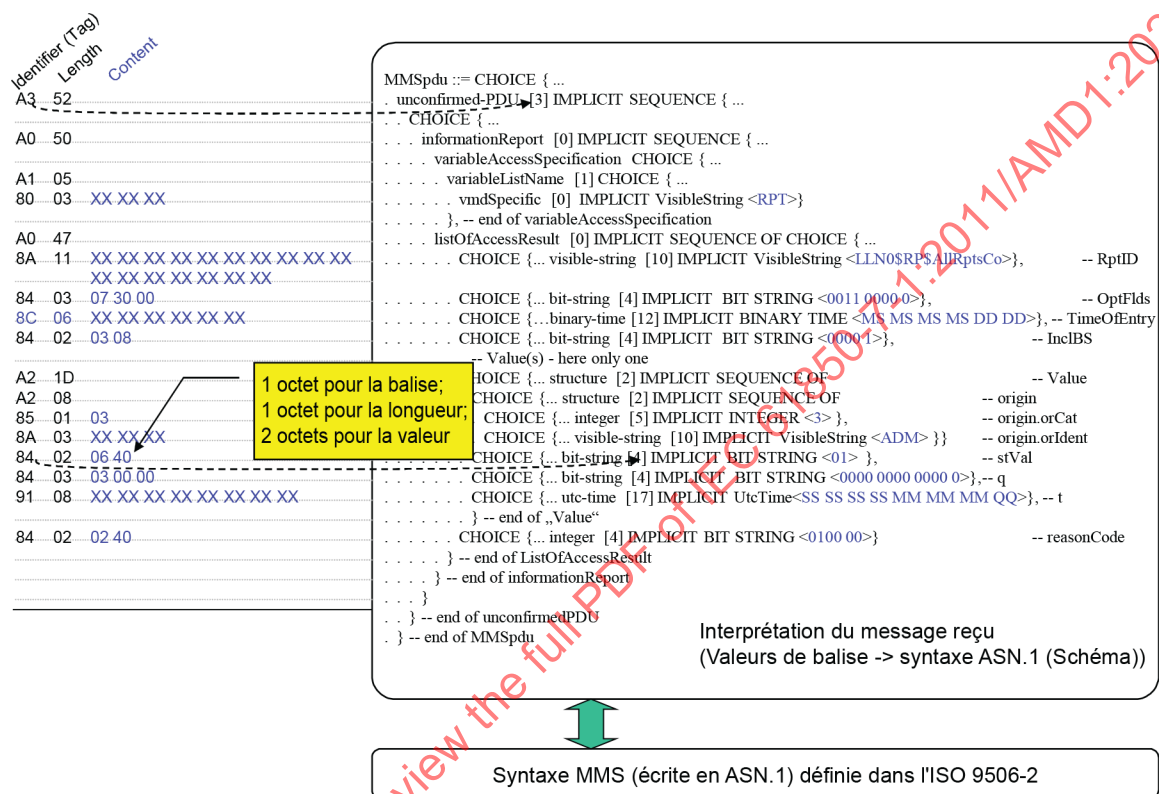


Figure F.9 – Message de rapport relatif aux informations de la MMS

Annexe G (normative)

Ingénierie des nœuds logiques LGOS/LSVS

Après l'Annexe F existante, ajouter les nouvelles annexes G à K suivantes:

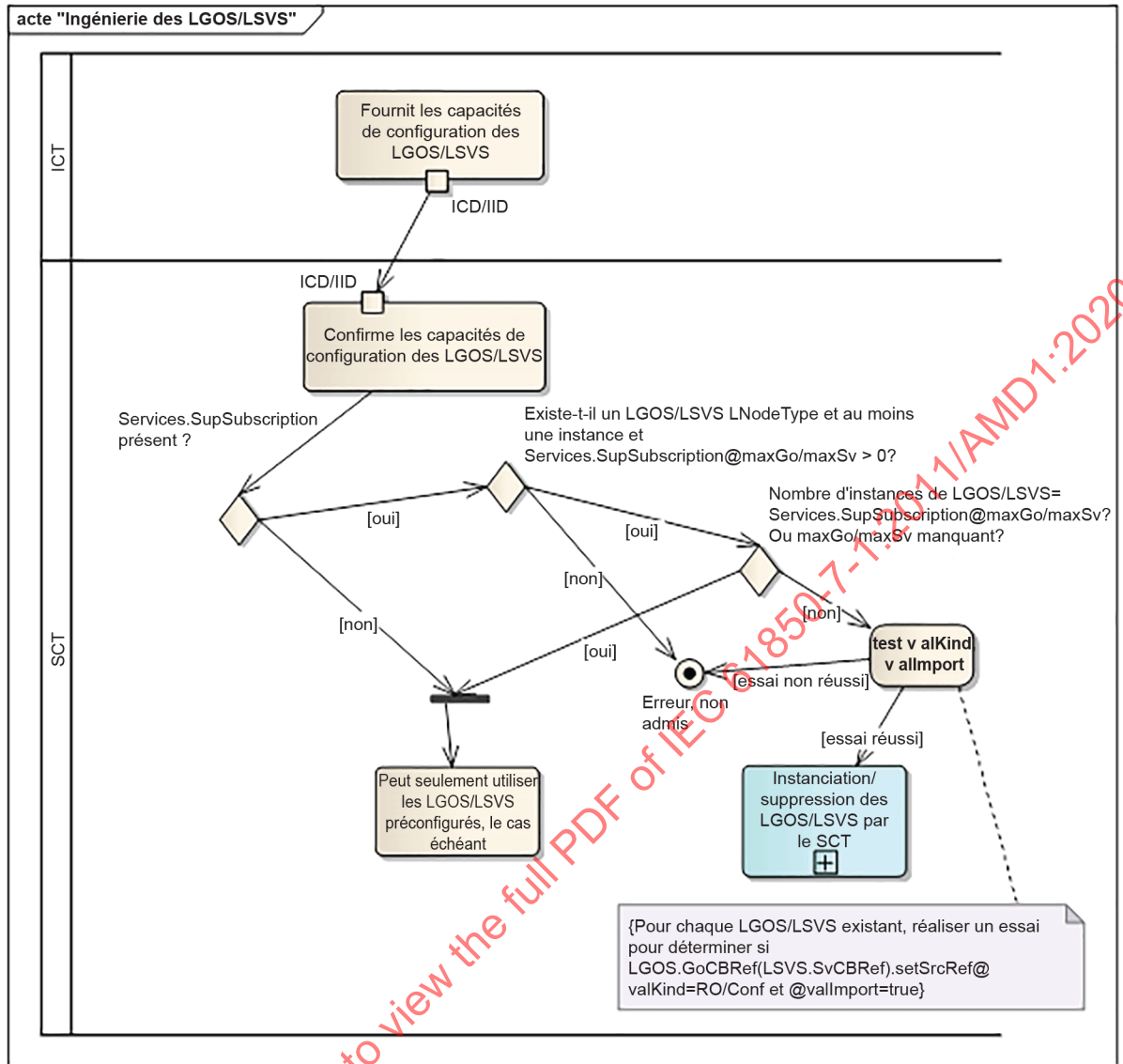
G.1 Généralités

La présente annexe définit les exigences de configuration des nœuds logiques utilisés pour surveiller les signaux GOOSE ou SMV. Lorsque la surveillance est configurée de manière statique, une instance de LGOS/LSVS par type de message GOOSE/SV reçu est nécessaire.

La configuration de ces nœuds logiques peut être effectuée par l'outil de configuration de l'IED (ICT) et/ou par l'outil de configuration du système (SCT). Les capacités de configuration des nœuds logiques de surveillance LGOS/LSVS, s'ils sont entièrement pris en charge, doivent être déclarées dans le fichier ICD/IID.

Pour obtenir une description complète de ces capacités ainsi qu'une description générale du processus d'ingénierie, voir l'IEC 61850-6.

Le diagramme de la Figure G.1 représente les conditions à remplir dans la description des capacités fournie par l'ICT afin que le SCT puisse ajouter ou supprimer les nœuds logiques de surveillance LGOS/LSVS.

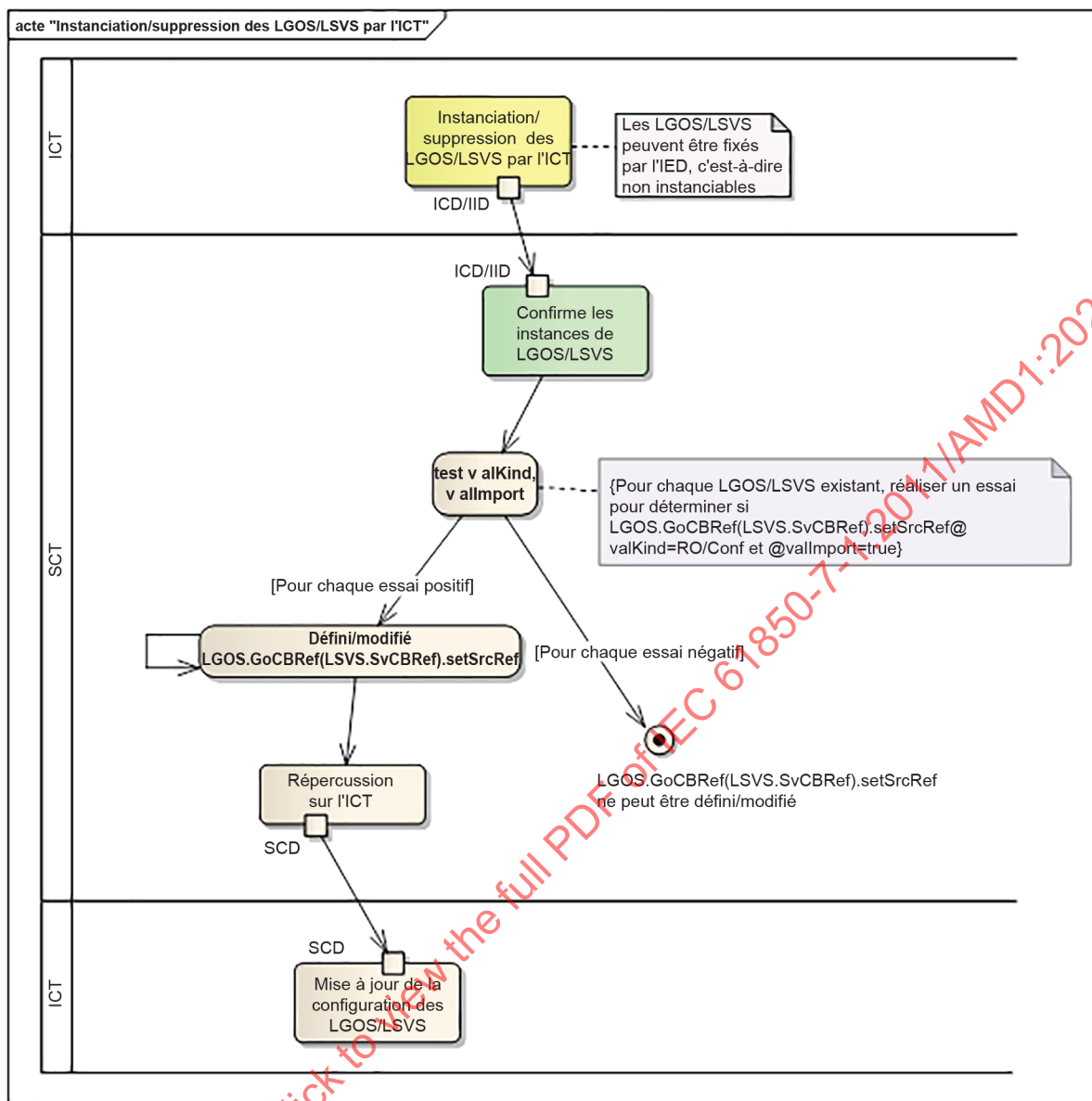


IEC

Figure G.1 – Ingénierie des nœuds logiques LGOS/LSVS

G.2 Ingénierie des nœuds logiques LGOS/LSVS par l'ICT

Le diagramme de la Figure G.2 représente le cas où l'instanciation/la suppression des nœuds logiques LGOS/LSVS est réalisée par l'ICT.



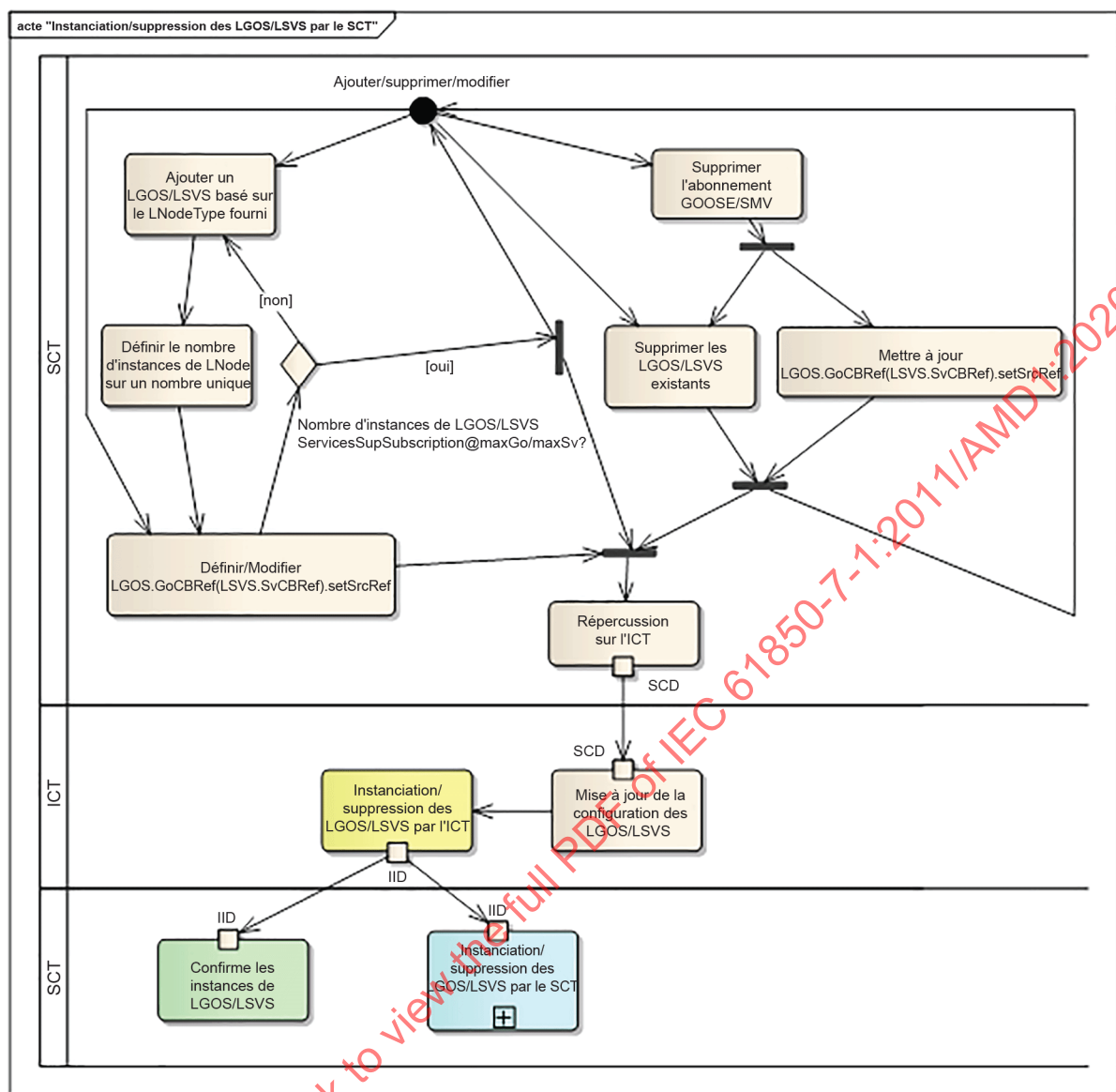
IEC

Figure G.2 – Ingénierie des nœuds logiques LGOS/LSVS par l'ICT

L'ICT peut procéder à l'instanciation/la suppression lors de la configuration ou l'IED fournit uniquement des instances fixes.

G.3 Ingénierie des nœuds logiques LGOS/LSVS par le SCT

Le diagramme de la Figure G.3 représente les activités relatives à l'ajout/la suppression des nœuds logiques de surveillance LGOS/LSVS par le SCT lorsque les conditions sont remplies dans la description des capacités fournie par l'ICT.



IEC

Figure G.3 – Ingénierie des nœuds logiques LGOS/LSVS par le SCT

Le flux de travail est le suivant:

- Le SCT ajoute/supprime les nœuds logiques LGOS/LSVS selon les besoins et le résultat est transmis à l'ICT par l'intermédiaire du fichier SCD.
- L'ICT peut ajouter/supprimer les nœuds logiques LGOS/LSVS si nécessaire et le résultat est renvoyé au SCT par l'intermédiaire du fichier IID.
- Le SCT peut ajouter ou supprimer les nœuds logiques LGOS/LSVS si nécessaire, ou confirmer la présence des nœuds logiques LGOS/LSVS et régler les valeurs des blocs de contrôle source à surveiller par un ou plusieurs nœuds logiques.

Les points suivants doivent être pris en compte:

- Le flux d'ingénierie permet à l'ICT et au SCT d'ajouter/de supprimer des instances de nœuds logiques de surveillance.
- Lors de la suppression d'un abonnement GOOSE/SMV, l'ICT ou le SCT peut soit supprimer l'instance de LGOS/LSVS correspondante, soit mettre à jour la référence au bloc de contrôle source en le vidant, ce qui implique que l'instance de LGOS/LSVS ne surveille aucun télégramme.

- Lors de la suppression d'une instance de LGOS/LSVS, l'ICT ne doit pas réaffecter un numéro aux instances restantes de LGOS/LSVS.

IECNORM.COM : Click to view the full PDF of IEC 61850-7-1:2011/AMD1:2020

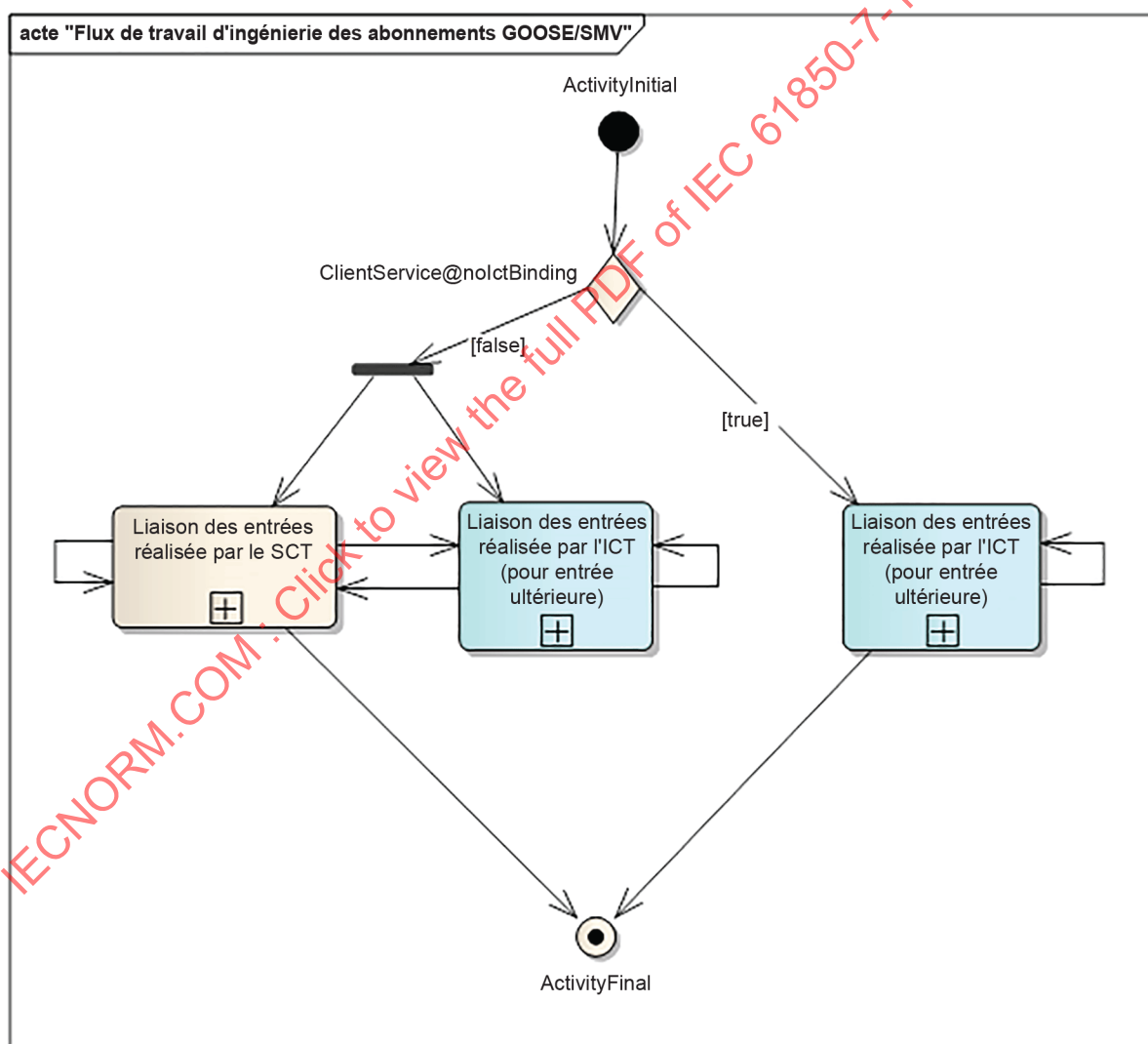
Annexe H (normative)

Configuration d'abonnement GOOSE/SMV

H.1 Généralités

Le langage SCL (IEC 61850-6) permet de recueillir des données sur le flux des signaux entre les IED au sein d'un système PUAS. Il permet également de gérer efficacement les entrées de signaux de communication qu'un client/abonné doit utiliser pour remplir ses fonctions. La présente annexe décrit le flux de travail minimal supposé être utilisé par les outils impliqués dans la configuration de l'abonnement GOOSE/SMV. Pour obtenir une description complète des activités impliquées dans les flux de travaux, voir l'IEC 61850-6.

La Figure H.1 représente deux flux de travaux en fonction des capacités déclarées par un outil.



IEC

Figure H.1 – Flux de travail d'ingénierie des abonnements GOOSE/SMV

Si l'outil IED (ICT) n'a aucune restriction et accepte que la liaison des entrées soit réalisée par l'outil système (SCT), le flux de travail "SCT supplied input binding" (liaison des entrées réalisée par SCT) s'applique. Sinon, le flux de travail "ICT supplied input binding" (liaison des entrées réalisée par ICT) doit s'appliquer. La capacité de prendre en charge les deux flux de travaux ou uniquement le flux de travail "ICT supplied input binding" est indiquée par l'attribut `noIctBinding` de l'élément `ClientServices` dans le fichier de description SCL de l'IED.

Quel que soit le flux de travail impliqué, la configuration des abonnements GOOSE/SMV doit être effectuée à l'aide de l'élément SCL `GSEControl.IEDName` (respectivement `SampledValueControl.IEDName`) dans le bloc de contrôle de l'éditeur. Côté abonné, l'élément SCL `Inputs` au niveau de l'élément du LN doit être utilisé pour indiquer quels membres `DataSet` doivent être utilisés à partir d'un message souscrit et par quel nœud logique.

H.2 Flux de travail "SCT supplied input binding" (liaison des entrées réalisée par le SCT)

La Figure H.2 représente le flux de travail prévu dans lequel l'ICT/IED accepte que le SCT réalise la liaison des entrées (`ClientServices@noIctBinding = FALSE`). Noter que le flux de travail ne montre pas le transfert des données de configuration vers l'IED.