INTERNATIONAL STANDARD

IEC 61850-7-1

First edition
2003-07

Communication networks and systems in substations –

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

Reference number
IEC 61850-7-1:2003(E)

Publication numbering

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

Consolidated editions

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

Further information on IEC publications

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

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

- **Catalogue of IEC publications**

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 On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

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

- **Customer Service Centre**

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

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

INTERNATIONAL STANDARD

IEC 61850-7-1

First edition
2003-07

Communication networks and systems in substations –

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

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

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XE

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMUNICATION NETWORKS AND SYSTEMS IN SUBSTATIONS –

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

FOREWORD

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

International Standard IEC 61850-7-1 has been prepared by IEC technical committee 57: Power system control and associated communications.

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

FDIS	Report on voting
57/637/FDIS	57/646/RVD

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

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

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

Part 1: Introduction and overview

Part 2: Glossary [1]

Part 3: General requirements

Part 4: System and project management

Part 5: Communication requirements for functions and device models

Part 6: Configuration description language for communication in electrical substations related to IEDs [2]

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

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

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

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

Part 8-1: Specific communication service mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-1 and ISO/IEC 9506-2) and to ISO/IEC 8802-3 [2]

Part 9-1: Specific communication service mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link

Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3 [2]

Part 10: Conformance testing [2]

The content of this part is based on existing or emerging standards and applications.

The committee has decided that the contents of this publication will remain unchanged until 2005. At this date, the publication will be

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

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

1 To be published.

2 Under consideration.

INTRODUCTION

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

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

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

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

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

Table 1 – Guide for the reader

User		**IEC 61850-1** (Introduction and overview)	**IEC 61850-5** (Requirements)	**IEC 61850-7-1** (Principles)	**IEC 61850-7-4** (Logical nodes and data classes)	**IEC 61850-7-3** (Common data classes)	**IEC 61850-7-2** (Information exchange)	**IEC 61850-6**[a] (Configuration language)	**IEC 61850-8-x IEC 61850-9-x**[a] (Concrete communication stack)
Utility	Manager	**x**	**–**	**Clause 5**	**–**	**–**	**–**	**–**	**–**
	Engineer	**x**	**x**	**x**	**x**	**x**	**In extracts**	**x**	**–**
Vendor	Application engineer	**x**	**x**	**x**	**x**	**x**	**In extracts**	**x**	**In extracts**
	Communication engineer	**x**	**x**	**x**	**–**	**–**	**x**	**–**	**x**
	Product manager	**x**	**x**	**x**	**x**	**In extracts**	**In extracts**	**In extracts**	**–**
	Marketing	**x**	**x**	**Clause 5**	**In extracts**	**In extracts**	**In extracts**	**In extracts**	**–**
Consultant	Application engineer	**x**	**x**	**x**	**x**	**x**	**–**	**x**	**–**
	Communication engineer	**x**	**–**	**x**	**–**	**–**	**x**	**x**	**x**
All others		**x**	**x**	**x**	**–**	**–**	**–**	**–**	**–**

The “**x**” means that this part of the IEC 61850 series should be read.

The “**in extracts**” means that extracts of this part of the IEC 61850 series should be read to understand the conceptual approach used.

The “**–**” means that this part of the IEC 61850 series may be read.

[a] These documents are under consideration.

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

Table 1 provides a simplified guide as to which parts of the IEC 61850 series should be read by various stakeholders. Four groups are shown: utility, vendor, various consultants, and others.

COMMUNICATION NETWORKS AND SYSTEMS IN SUBSTATIONS –

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

1 Scope

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

– substation-specific information models for substation automation systems,

– device functions used for substation automation purposes, and

– communication systems to provide interoperability within substations.

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

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

– substation to substation information exchange,

– substation to control centre information exchange,

– information exchange for distributed automation,

– information exchange for metering,

– condition monitoring and diagnosis, and

– information exchange with engineering systems for device configuration.

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

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

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

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

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61850-2, *Communication networks and systems in substations – Part 2: Glossary* [3]

IEC 61850-5, *Communication networks and systems in substations – Part 5: Communication requirements for functions and devices models* [3]

IEC 61850-7-2, *Communication networks and systems in substations – Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)*

IEC 61850-7-3, *Communication networks and systems in substations – Part 7-3: Basic communication structure for substation and feeder equipment – Common data classes*

IEC 61850-7-4, *Communication networks and systems in substations – Part 7-4: Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes*

ISO/IEC 8802-3:2000, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

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

ISO 9506-1:2003, *Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*

ISO 9506-2:2003, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

3 Terms and definitions

For the purposes of this International Standard, the terms and definitions given in IEC 61850-2[3] as well as the following, apply.

3.1
information
knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning

(IEV 101-12-01)

3.2
information model
represents the knowledge concerning substation functions and devices in which the functions are implemented. This knowledge is made visible and accessible through the means of the IEC 61850 series. The model describes in an abstract way a communication oriented representation of a real function or device.

3 To be published.

3.3
model
a representation of some aspect of reality. The purpose of creating a model is to help understand, describe, or predict how things work in the real world by exploring a simplified representation of a particular entity or phenomenon. The focus of the model defined in IEC 61850-7-x is on the communication features of the data and functions modelled.

4 Abbreviated terms

ACSI	Abstract Communication Service Interface
ASN.1	Abstract Syntax Notation One
API	Application Program Interface
CDC	Common Data Class
CT	Current Transformer
IED	Intelligent Electronic Device
LD	Logical Device
LN	Logical Node
LLN0	Logical Node Zero
LPHD	Logical Node Physical Device
MMS	Manufacturing Message Specification
PHD	Physical Device
PICOM	Piece Of Communication
SCSM	Specific Communication Service Mapping
SoE	Sequence Of Events
UML	Unified Modelling Language
VMD	Virtual Manufacturing Device
VT	Voltage Transformer
XML	eXtended Markup Language

5 Overview of concepts the IEC 61850 series

5.1 Objective

IEC 61850-7-4, IEC 61850-7-3, IEC 61850-7-2, IEC 61850-6, and IEC 61850-8-1 are closely related. This Subclause provides an overview of these parts and it describes how these parts are interwoven.

Each part defines a specific aspect of a substation IED:

– IEC 61850-7-4 defines specific information models for substation automation functions (for example, breaker with status of breaker position, settings for a protection function, etc.) – what is modelled and could be exchanged,
– IEC 61850-7-3 has a list of commonly used information (for example, for double point control, 3-phase measurand value, etc.) – what the common basic information is,
– IEC 61850-7-2 provides the services to exchange information for the different kinds of functions (for example, control, report, get and set, etc.) – how to exchange information,

– IEC 61850-6 offers the formal configuration description of a substation IED including the description of the relations with other IEDs and with the power process (single line diagram) – how to describe the configuration, and

– IEC 61850-8-1 defines the concrete means to communicate the information between IEDs (for example, the application layer, the encoding, etc.) – how to serialise the information during the exchange.

5.2 Topology and communication functions of substation automation systems

As shown by the topology in Figure 1, one focus of the IEC 61850 series is the support of substation automation functions by the communication of (numbers in brackets refer to the figure):

– sampled value exchange for CTs and VTs (1),

– fast exchange of I/O data for protection and control (2),

– control and trip signals (3),

– engineering and configuration (4),

– monitoring and supervision (5),

– control-center communication (6),

– time-synchronisation,

– etc.

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

Many functions are implemented in intelligent electronic devices (IED); various IEDs are shown in Figure 1. Several functions may be implemented in a single IED or one function may be implemented in one IED and another function may be hosted by another IED. IEDs (i.e., the functions residing in IEDs) communicate with functions in other IEDs by the information exchange mechanisms of this standard. Therefore, functions distributed over more than one IED may be also implemented.

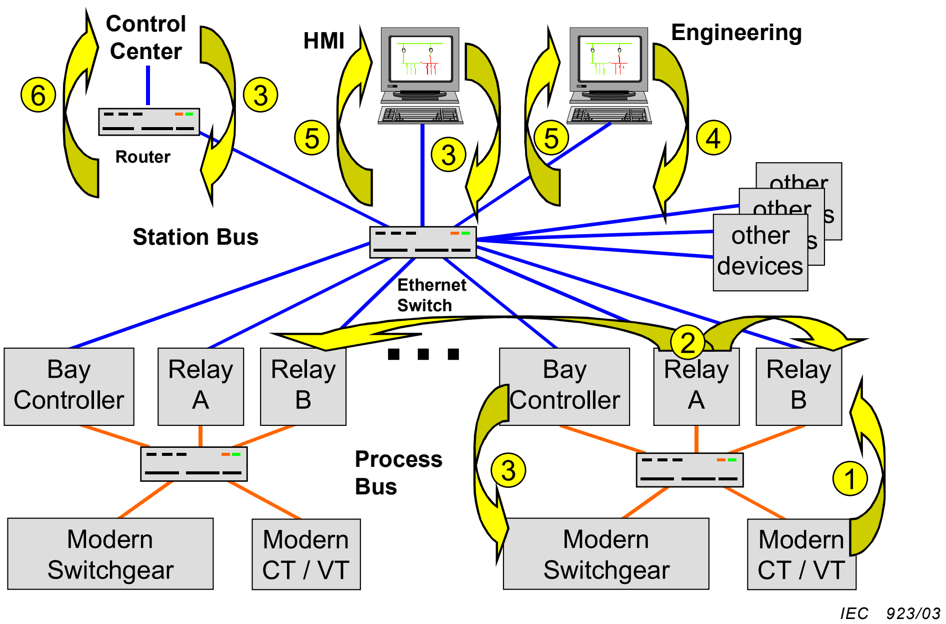


Figure 1 – Sample substation automation topology

5.3 The information models of substation automation systems

The information exchange mechanisms rely primarily on well defined information models. These information models and the modelling methods are at the core of the IEC 61850 series. The IEC 61850 series uses the approach to model the common information found in real devices as depicted in Figure 2. All information made available to be exchanged with other devices is defined in the standard. The model provides for the substation automation system an image of the analogue world (power system process, switchgear).

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

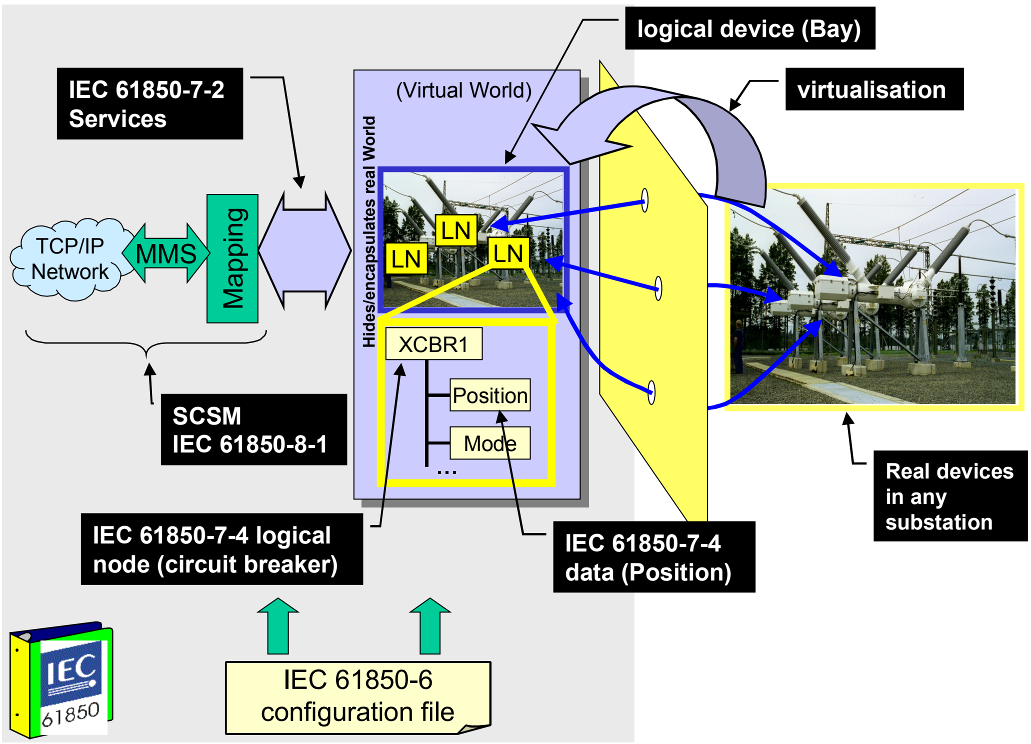


Figure 2 – Modelling approach (conceptual)

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

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

Real devices on the right hand side of Figure 2 are modelled as a virtual model in the middle of the figure. The logical nodes defined in the logical device (for example, bay) correspond to well known functions in the real devices. In this example the logical node **XCBR** represents a specific circuit breaker of the bay to the right.

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

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

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

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

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

Table 2 lists all groups of logical nodes defined in IEC 61850-7-4. About 90 logical nodes covering the most common applications of substation and feeder equipment are defined. One main focus is the definition of information models for protection and protection related applications (38 logical nodes out of 88). These two groups comprise nearly half of the logical nodes. This impression results from the very dedicated definition of protection functions in history because of the high importance of protection for safe and reliable operation of the power system.

NOTE Some attention is given to control functions which historically have not been defined in such a granularity since they represent a few very common and also important tasks.

The importance of monitoring functions is increasing.

Table 2 – LN groups

Logical node groups	Number of logical nodes
System logical nodes	3
Protection functions	28
Protection related functions	10
Supervisory control	5
Generic references	3
Interfacing and archiving	4
Automatic control	4
Metering and measurement	8
Sensors and monitoring	4
Switchgear	2
Instrument transformer	2
Power transformer	4
Further power system equipment	15
Total number of logical nodes	92

IEC 61850 has well-defined rules to define additional logical nodes and data, for example, for additional functions within substations or for other application domains such as wind power plants. For details on the extension rules, see Clause 14 of this standard and the Annex A of IEC 61850-7-4.

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

– Distance protection
– Differential protection
– Overcurrent
– Undervoltage
– Directional over power
– Volts per Hz relay
– Transient earth fault
– Directional element
– Harmonic restraint
– Protection scheme
– Zero speed or underspeed
– ...
– Measurement
– Metering
– Sequence and imbalance
– Harmonics and interharmonics
– Differential measurements
– ...
– Switch control
– Circuit breaker
– Circuit switch
– ...

Most logical nodes provide information that can be categorised as depicted in Figure 3. The semantic of a logical node is represented by data and data attributes. Logical nodes may provide a few or up to 30 data. Data may contain a few or even more than 20 data attributes. Logical nodes may contain more than 100 individual information (points) organised in a hierarchical structure.

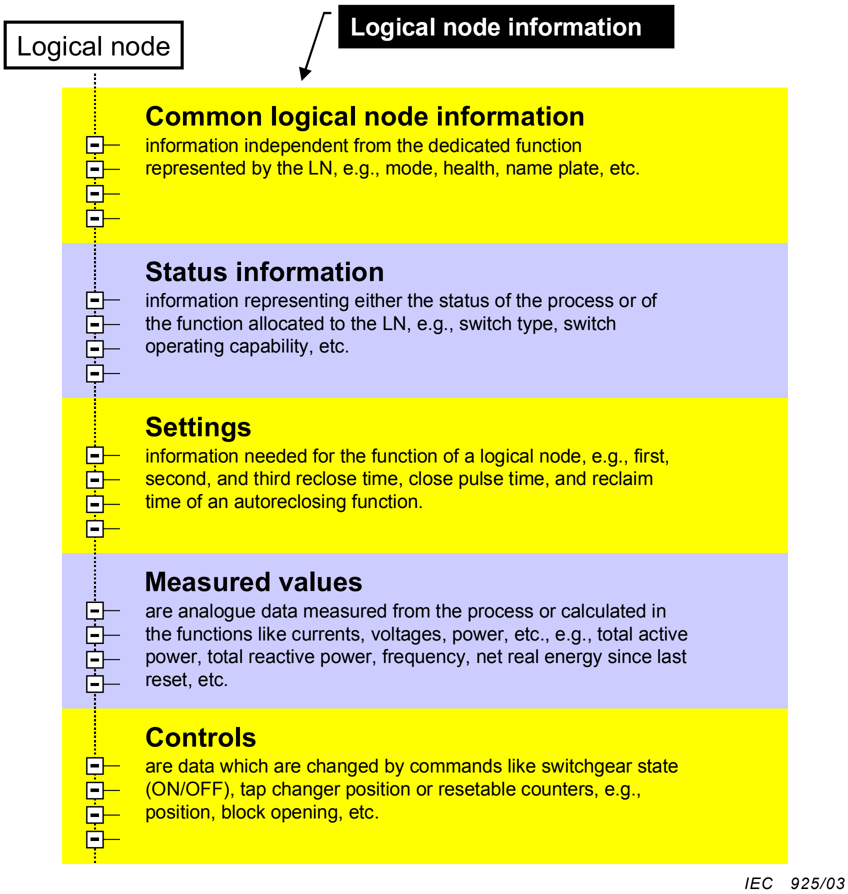


Figure 3 – Logical node information categories

IEDs are built up by composing logical nodes as depicted in Figure 4. The logical nodes are the building blocks of substation IEDs, for example, circuit breaker (**XCBR**) and others. In the example for each phase, one instance of XCBR is used.

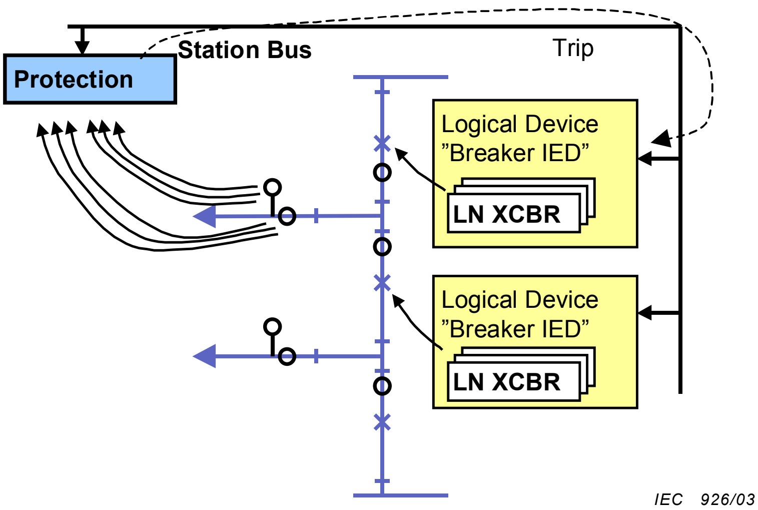


Figure 4 – Build up of devices (principle)

In Figure 4, the protection IED receives the values for the voltage and current from conventional VT and CT. The protection functions in the protection device may detect a fault and issue or send a trip signal via the station bus. The standard supports also IEDs for non-conventional VTs and CTs sending voltage and current as samples to the protection over a serial link.

The **logical nodes** are used to build up substation IEDs.

5.5 The semantic is attached to data

The mean number of specific data provided by logical nodes defined in IEC 61850-7-4 is approximately 20. Each of the data (for example, position of a circuit breaker) comprises several details (the data attributes). The position (named "**Pos**") of a circuit breaker is defined in the logical node **XCBR** (see Figure 5). The position is defined as data. The category of the position in the logical node is "controls" – the position can be controlled via a control service.

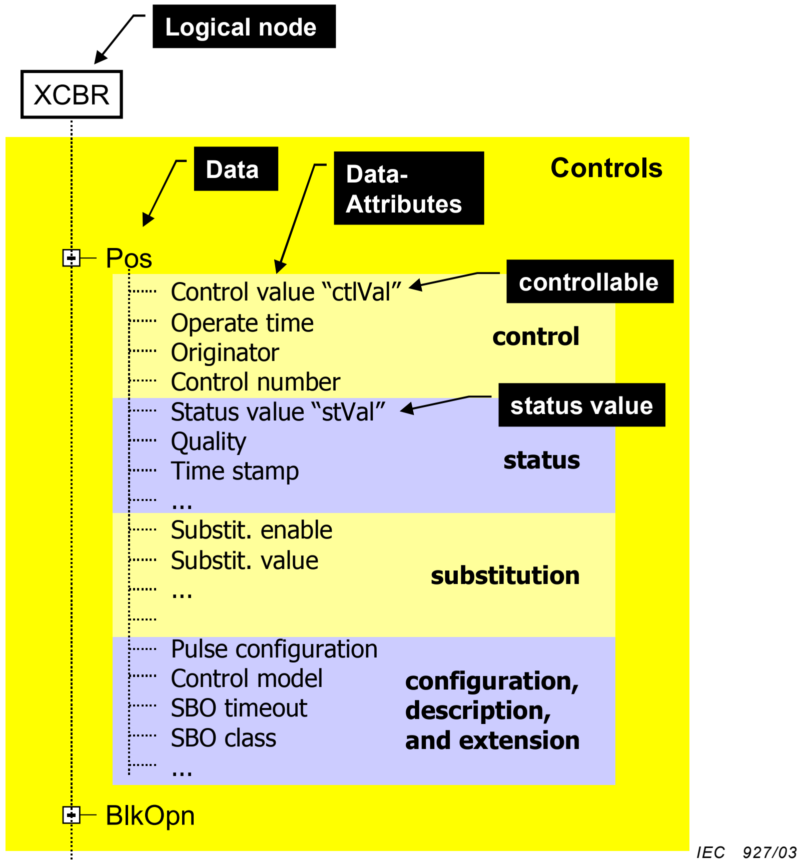


Figure 5 – Position information depicted as a tree (conceptual)

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

– control (status, measured/metered values, or settings),

– substitution,

– configuration, description and extension.

The data example **Pos** has approximately 20 data attributes. The data attribute **Pos.ctlVal** represents the controllable information (can be set to ON or OFF). The data attribute **Pos.stVal** represents the position of the real breaker (could be in intermediate-state, off, on, or bad-state).

The position also has information about when to process the control command (**Operate time**), the originator that issued the command, and the control number (given by the originator in the request). The quality and time stamp information indicate the current validity of the status value and the time of the last change of the status value.

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

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

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

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

Attribute name	Attribute type	FC	TrgOp	Value/value range	M/O/C
ctlVal	BOOLEAN	CO		off (FALSE) | on (TRUE)	AC_CO_M
stVal	CODED ENUM	ST	dchg	intermediate-state | off | on | bad-state	M

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

– the services allowed: functional constraint -> FC=CO means that specific services can be applied only (for example CO refers to the control service),
– the trigger conditions that cause a report to be sent: TrgOp=dchg means that a change in the value of that attribute causes a report,
– the value or value range,
– the indication if the attribute is optional (O), mandatory (M), conditional mandatory (X_X_M), or conditional optional (X_X_O). The conditions result from the fact that not all attributes are independent from each other.

The data attribute names are standardised (i.e., these 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 at the end of IEC 61850-7-3; for example:

Data attribute name	Semantics
ctlVal	Determines the control activity.
stVal	Status value of the data.

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

The position information **Pos** as shown in Figure 5 has many data attributes that can found in many other switching-specific applications. The prime characteristic of the position is the data attribute **stVal** (status value) which represents four states: intermediate-state | off | on | bad-state. These four states (represented usually with two bits) are commonly known as “double point” information. The whole set of all the data attributes defined for the data **Pos** (position) is called a “common data class” (CDC). The name of the common data class of the double point information is **DPC** (controllable double point).

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

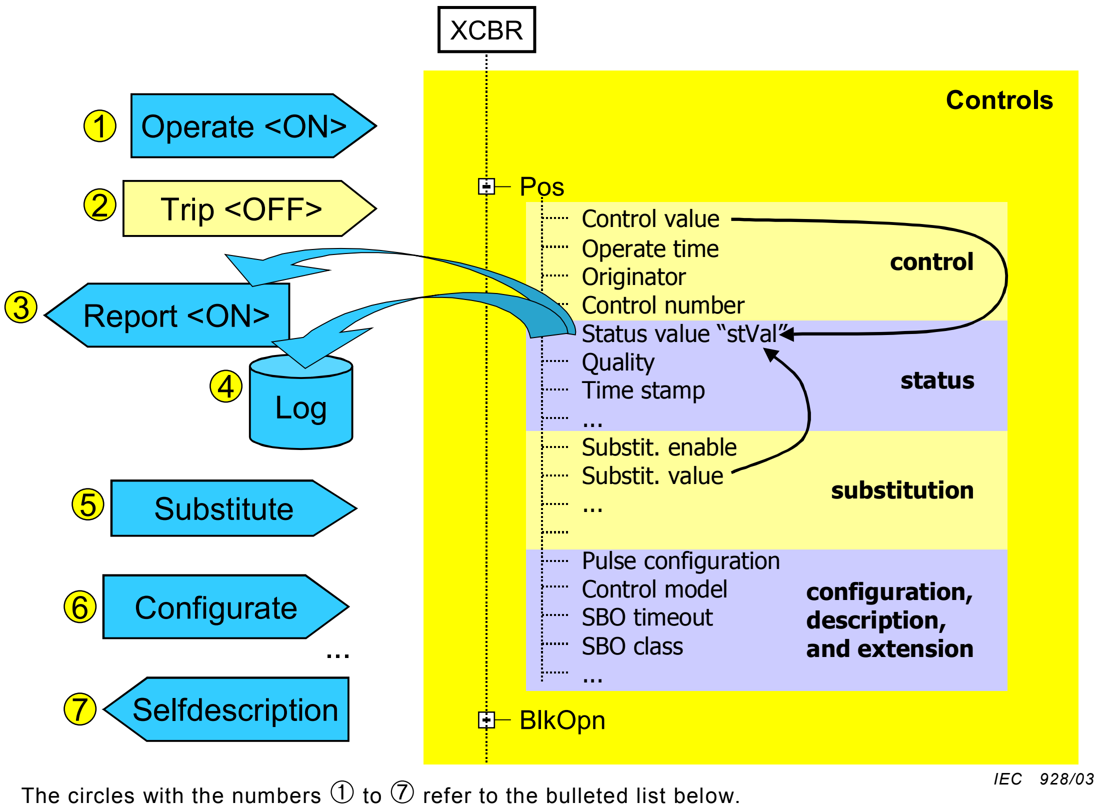
IEC 61850-7-3 defines common data classes for a wide range of well known applications. The core common data classes are classified into the following groups:

– status information,
– measurand information,

– controllable status information,
– controllable analogue information,
– status settings,
– analogue settings, and
– description information.

5.6 The services to exchange information

The logical nodes, data, and data attributes are defined mainly to specify the information required to perform an application, and for the exchange of information between IEDs. The information exchange is defined by means of services. An excerpt of the services is displayed in Figure 6.



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

Figure 6 – Service excerpt

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

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

- control devices (operate service or by multicast trip signals) (see Figure 6, ①),
- fast and reliable peer-to-peer exchange of status information (tripping or blocking of functions or devices) (see Figure 6, ②),
- reporting of any set of data (data attributes), SoE – cyclic and event triggered (see Figure 6, ③),
- logging and retrieving of any set of data (data attributes) – cyclic and event triggered (see Figure 6, ④),
- substitution (see Figure 6, ⑤),

- handling and setting of parameter setting groups,
- transmission of sampled values from sensors,
- time synchronisation,
- file transfer,
- online configuration (see Figure 6, ⑥), and
- retrieving the self-description of a device (see Figure 6, ⑦).

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

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

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

5.7 Services mapped to concrete communication protocols

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

The specific syntax (format) and especially the encoding of the messages that carry the service parameters of a service and how these are passed through a network are defined in a specific communication service mapping (SCSM). One SCSM – IEC 61850-8-1 – is the mapping of the services to MMS (ISO 9506) and other provisions like TCP/IP and Ethernet (see Figure 7) other ones are IEC 61850-9-1 and IEC 61850-9-2.

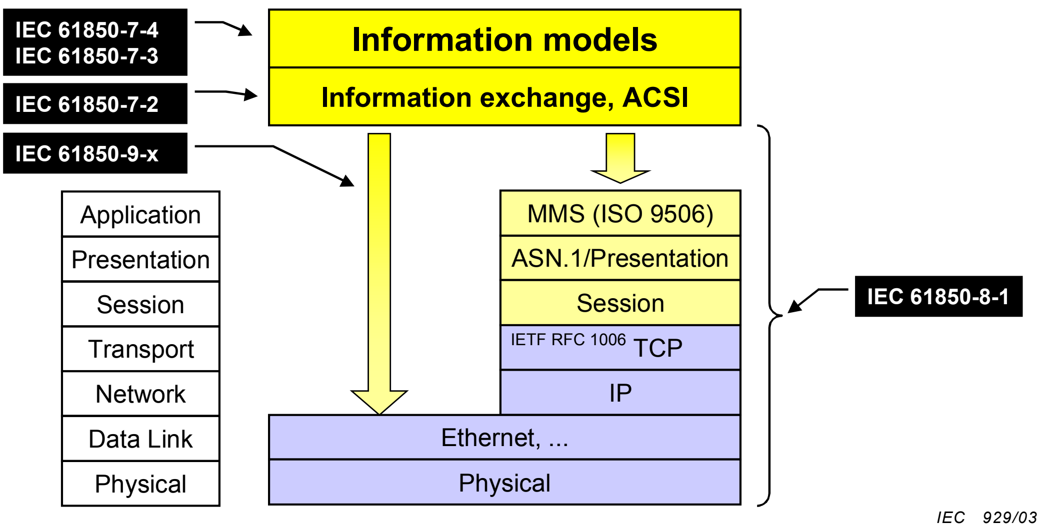


Figure 7 – Example of communication mapping

Additional mappings to other communication stacks are possible. The ACSI is independent of the mappings.

5.8 The configuration of a substation

The logical nodes, data, and data attributes as well as the services used and concrete communication means provided by a physical IED must be configured. The configuration contains the formal description of the various objects and the relations between these objects and the concrete substation equipment (switchyard). At the application level the switchyard topology itself and the relation between the switchyard structure and the SAS functions (the corresponding logical nodes, data and data attributes configured in the IEDs) are described.

IEC 61850-6 specifies a description language for configurations of electrical substation IEDs. This language is called substation configuration description language (SCL).

The substation configuration contains a static view of the complete substation. The configuration may be used for describing re-usable parts or for complete IEDs that can be employed immediately:

– pre-configured IEDs with a fixed number of logical nodes based on a function library, but with no binding to a specific process;
– pre-configured IEDs with a pre-configured semantic for a process part of a certain structure, for example a double busbar GIS line feeder;
– complete process configuration with all IEDs bound to individual process functions and primary equipment, enhanced by the access control object definitions (access allowances) for all possible communication partners;
– ready to run IED with all communication links ready to run. This is required if an IED is not capable dynamically opening connections;

The configuration language is based on the XML schema language.

5.9 Summary

Figure 8 exhibits a summary of Clause 5. The four main building blocks are

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

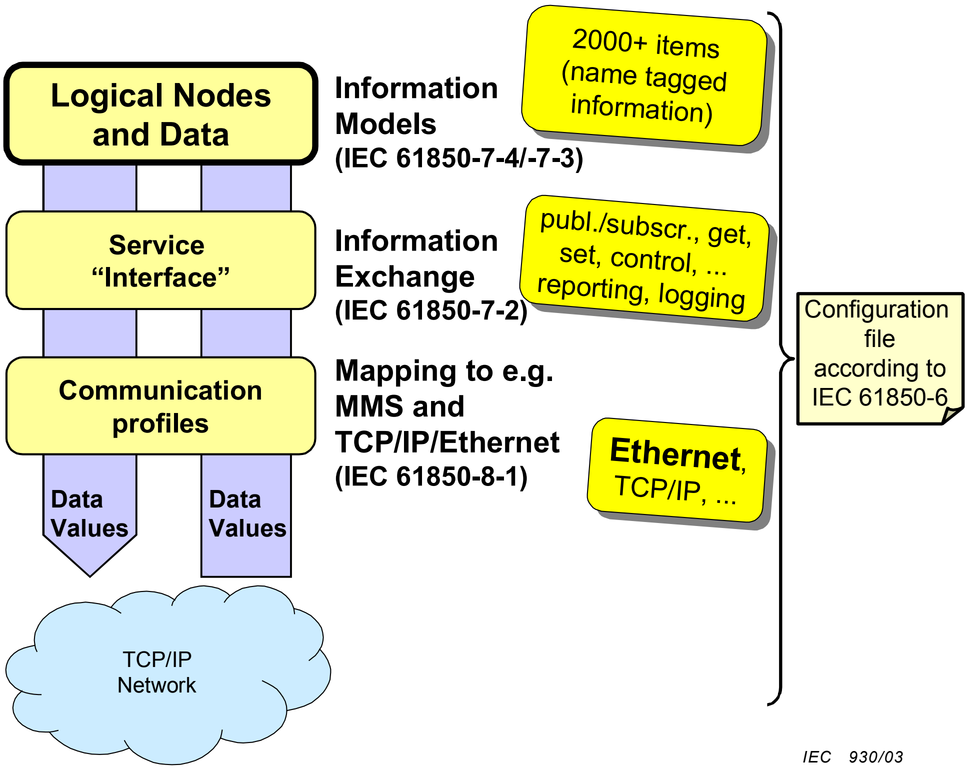


Figure 8 – Summary

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

The information is separated from the presentation and from the information exchange services.

The information exchange services are separated from the concrete communication profiles.

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

6 Modelling approach of the IEC 61850 series

6.1 Decomposition of application functions and information

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

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

The decomposition process (to get the most common logical nodes) and the composition process (to build up devices using logical nodes) are depicted in Figure 9. The data classes contained in logical nodes have been defined to support the most common applications in an understandable and commonly accepted way.

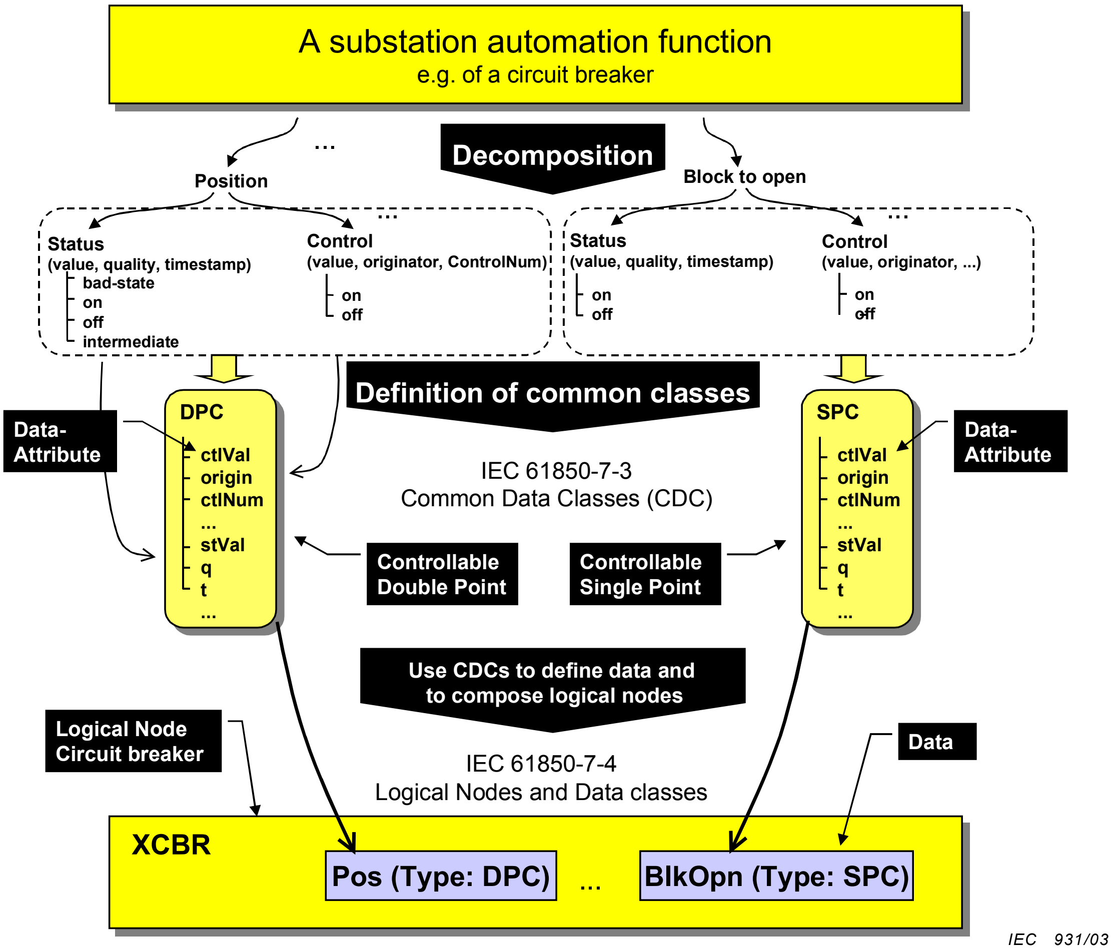


Figure 9 – Decomposition and composition process (conceptual)

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

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

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

IEC 61850-7-3 defines some 30 common data classes for status, measurands, controllable status, controllable analogue, status settings, and analogue settings.

6.2 Creating information models by stepwise composition

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

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

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

In the next step, the common data classes are used to define the (substation domain-specific) data classes (see lower half of Figure 9). These data classes (defined in IEC 61850-7-4) are specialised common data classes, for example, the data class **Pos** (a specialisation of **DPC**) inherits all data attributes of the corresponding common data class **DPC**, i.e., the **ctlVal**, **origin**, **ctlNum**, etc. The semantic of the class **Pos** is defined at the end of IEC 61850-7-4.

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

IEC 61850-7-4 defines some 90 logical nodes making use of the some 450 data classes. The logical node **XCBR** comprises about 20 data classes. A brief description of the logical node **XCBR** is given in Table 3.

Table 3 – Logical node class XCBR (conceptual)

Common Logical Node Information
Mode
Behaviour
Health
Name plate
Optional Logical Node Information
Local operation
External equipment health
External equipment name plate
Operation counter resetable
Operation counter
Operation time
Local operation (local means without substation automation communication, hardwired direct control)
Operation counter
External equipment health
External equipment name plate
Controls
Switch position (see below for details)
Block opening
Block closing
Charger motor enabled
Metered Values
Sum of Switched Amperes, resetable
Status Information
Circuit breaker operating capability
Point On Wave switching capability
Circuit breaker operating capability when fully charged

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

The content of the marked "switch position" (name = **Pos**) is introduced in Figure 10.

IEC 61850-7-x uses tables for the definition of the logical node classes and data classes (IEC 61850-7-4), the common data classes (IEC 61850-7-3) and service models (IEC 61850-7-2). Data classes and data attributes form a hierarchical structure as depicted in Figure 10. The data attributes of the data class **Pos** are organised in a way that all attributes for control (status, substitution, configuration, etc.) are listed together.

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

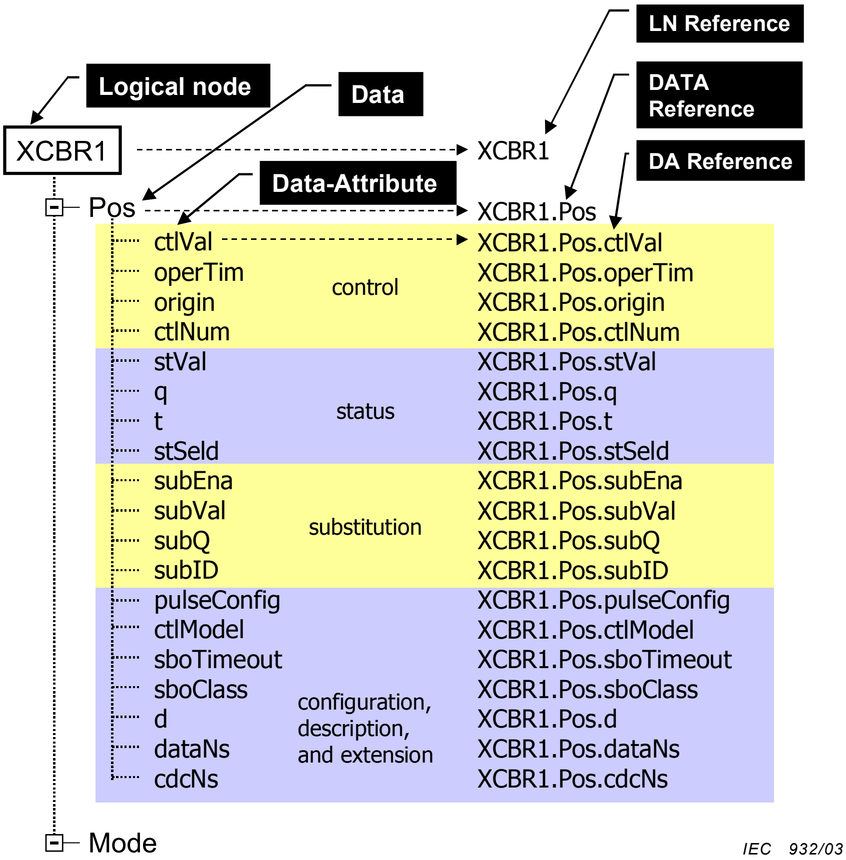


Figure 10 – XCBR1 information depicted as a tree

The instance **XCBR1** (the first instance of **XCBR**) is the root at the level of logical nodes. The object reference **XCBR1** references the complete tree below. The **XCBR1** contains data, for example, **Pos** and **Mode**. The data **Pos** (position) is precisely defined in IEC 61850-7-4 (see excerpt of the description):

Description of data

Data Name	Semantics
...	...
Pos	This data is accessed when performing a switch command or to verify the switch status or position. When this data is also used for a hand-operated switch, the (optional) CtlVal attribute in IEC 61850-7-3 does not exist.
...	...

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

The information exchange services that access the data attributes make use of the hierarchical tree. The controllable data attribute is defined with **XCBR1.Pos.ctlVal**. The control service operates on exactly that controllable data attribute of the circuit breaker. The status information could be referenced as a member (**XCBR1.Pos.stVal**) of a data set named "AlarmXCBR". The data set could be referenced by a reporting control block named

“Alarm”. The report control block could be configured to send a report to a specific computer each time a circuit breaker changes its state (from open to close or from close to open).

6.3 Example of an IED composition

Figure 11 shows examples of different logical nodes composed into an IEDs. The logical nodes involved are **PTOC** (time overcurrent protection), **PDIS** (distance protection), **PTRC** (trip conditioning) and **XCBR** (circuit breaker). Case 1 shows a protection device with two functions, which are hardwired with the circuit breaker. Case 2 shows a protection device with two functions where the trip is communicated via a trip message over a network to the circuit breaker LN. Case 3 shows the two protection functions in dedicated devices, which may operate both in a fault and where the trips are transmitted as trip messages via the network independently to the circuit breaker LN (**XCBR**).

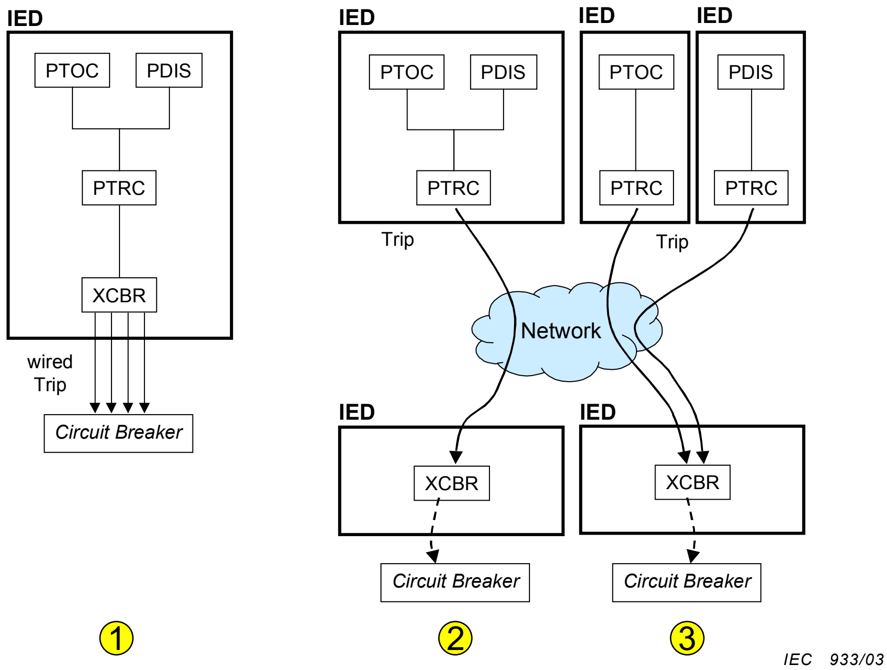


Figure 11 – Example of IED composition

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

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

6.4 Information exchange models

6.4.1 Introduction

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

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

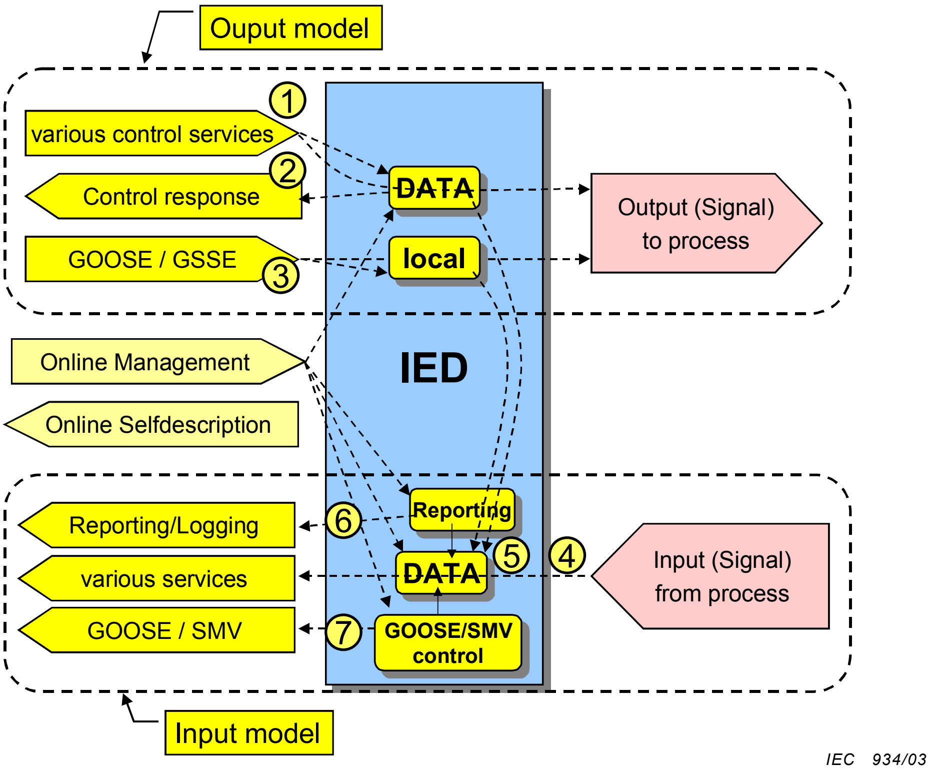
Several services are defined for each model. The services operate on data, data attributes, and other attributes usually contained in logical nodes. The numbers in the circles in

Figure 12 are used in 6.4.2 and Figures 13, 14, 15, 17, 19 and 21 as references for the description.

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

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

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



The numbers in the circles in this figure are used in 6.4.2 and Figures 13, 14, 15, 17, 19 and 21 as references for the description.

Figure 12 – Output and Input model (principle)

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

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

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

6.4.2 Output model

6.4.2.1 Control model concept

The concept of the control model is depicted in Figure 13. The example is a circuit breaker logical node (**XCBR**) with the data attribute **XCBR.Pos.ctlVal** (shown in Figure 14). Before the control service request performs the change of the position of a real device, some conditions have to be met, for example, the output can be generated only if the local/remote switch is in the "remote" position and the interlocking node (**CILO**) has released this operation. The following chain of conditions to be met may possibly include:

– local/remote switch of the circuit breaker **XCBR.Loc**,
– mode information of the circuit breaker **XCBR.Mod**,
– check conditions of the device, and
– other attributes of the controllable date, for example, interlocking, pulse configuration, control model, sbo class, and sbo timeout as defined in the common data class **DPC** (controllable double point in IEC 61850-7-3.

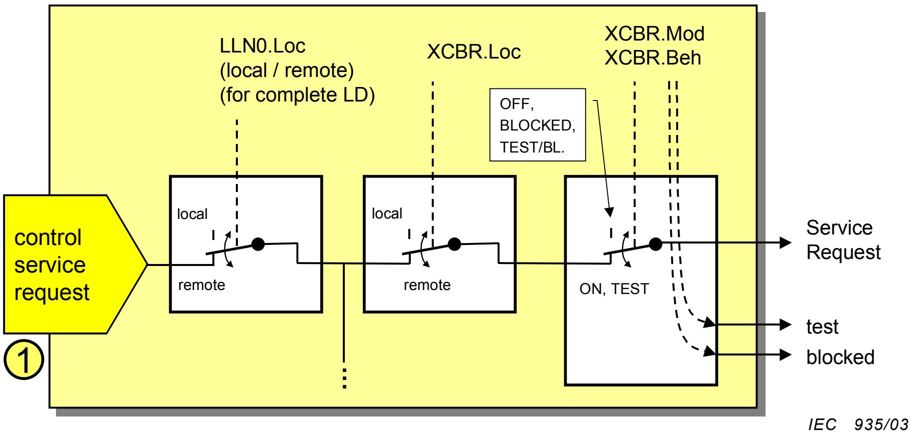


Figure 13 – Output model (step 1) (conceptual)

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

The output signal may be issued over a wired interface to the circuit breaker or may be communicated over a bus interface.

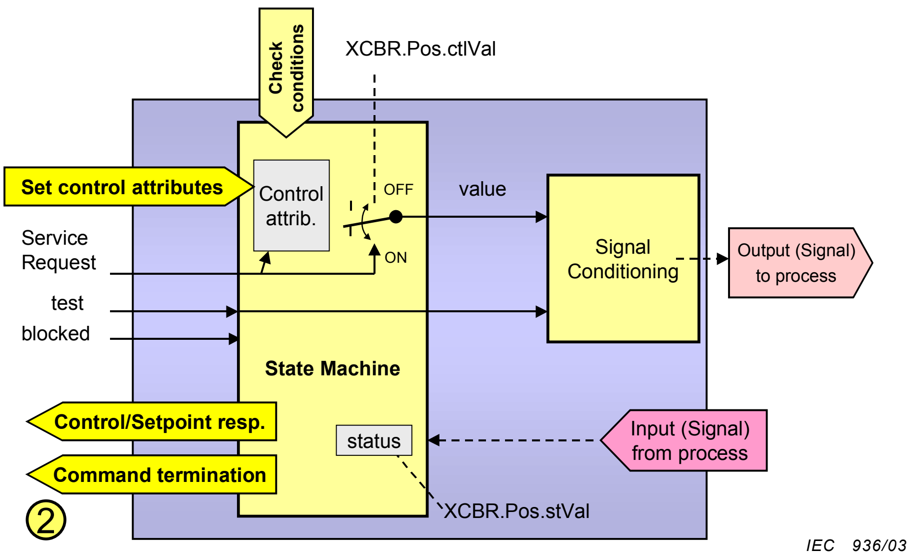


Figure 14 – Output model (step 2) (conceptual)

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

6.4.2.2 GSE model concept

The generic substation event (GSE – GOOSE and GSSE) provides the peer-to-peer information exchange between the input data values of one IED to the output data of many other IEDs (multicast). The GOOSE and GSSE messages received by an IED may be used to compute data for internal purposes also. An example for internal purposes are received switch positions to calculate the interlocking conditions locally.

NOTE 1 The GOOSE and GSSE data values are defined in the input model described in 6.4.3.

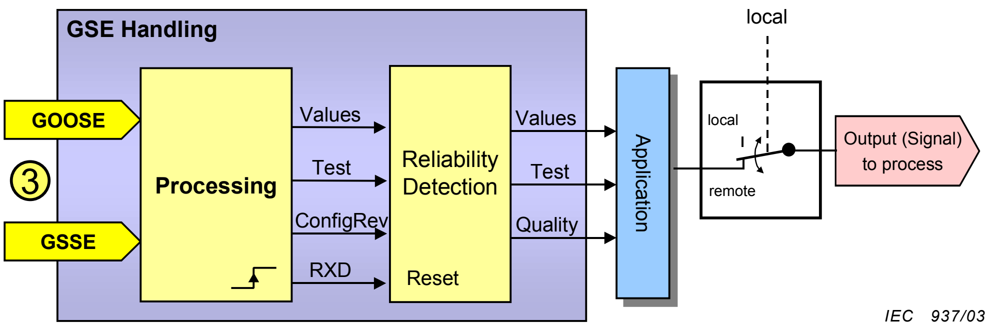


Figure 15 – GSE output model (conceptual)

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

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

6.4.2.3 Attributes of data and control blocks

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

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

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

All control block attributes can be read by another IED.

6.4.2.4 Setting data and setting group control block

A special treatment of output data values is required for setting data contained in several logical nodes as defined in IEC 61850-7-4, for example, the settings for the voltage controlled overcurrent protection logical node **PVOC** (see Figure 16). The setting data (for example **AVCrv**, **TmACrv**, **TmMult**, etc.) have as many values as setting groups are defined. Each setting group has a consistent set of values.

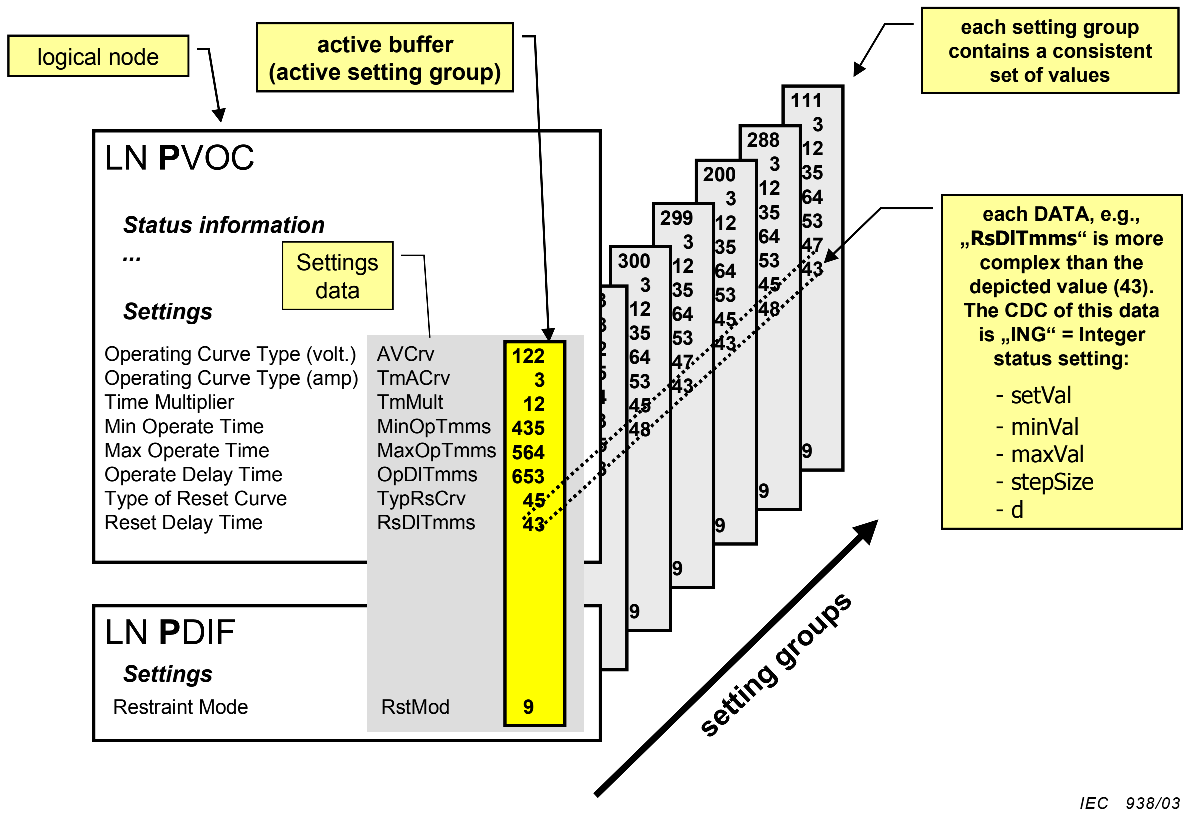


Figure 16 – Setting data (conceptual)

The values depicted are complex in the sense that each data has a type derived from a common data class. The **RsDlTmms** is derived from the common data class **ING**. The ING has several data attributes as listed in Table 4.

Table 4 – Excerpt of integer status setting

ING class					
Attribute name	**Attribute type**	**FC**	**TrgOp**	**Value/value range**	**M/O/C**
...	...				
DataAttribute					
				setting	
setVal	INT32	SP			AC_NSG_M
setVal	INT32	SG, SE			AC_SG_M
configuration, description and extension					
minVal	INT32	CF			O
maxVal	INT32	CF			O
stepSize	INT32U	CF		1 ... (maxVal – minVal)	O
d	VISIBLE STRING255	DC		Text	O
...	...	...			...

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

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

6.4.3 Input model

6.4.3.1 Input analogue signal acquisition

The concept of the input analogue signal acquisition is depicted in Figure 17. Normally, the raw signal is conditioned by a signal conditioner. For our model, an analogue input exists as data not before it is converted from analogue to digital. The sample rate (data attribute smpRate of a configurable data) determines how often the value shall be sampled. The conditions to be met before the value can be communicated (modelled as the data attribute instMag of the data, for example, a voltage of a specific phase – see Figure 18) may comprise the values of the following attributes:

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

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

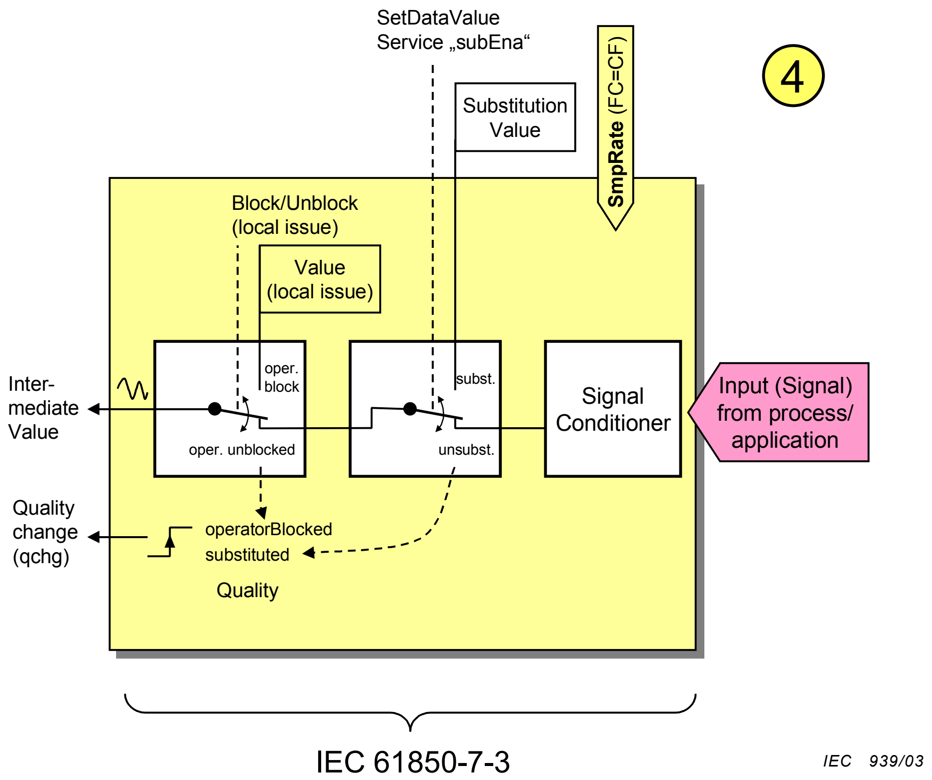


Figure 17 – Input model for analogue values (step 1) (conceptual)

6.4.3.2 Data attribute value processing, monitoring and event detection

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

The second application is the calculation of the deadbanded value, the **mag** value. The deadbanded value shall be based on a deadband calculation from **instMag** as illustrated in Figure 18. The value of **mag** shall be updated to the current value of **instMag** when the value has changed according the value of the configuration parameter **db** of this data.

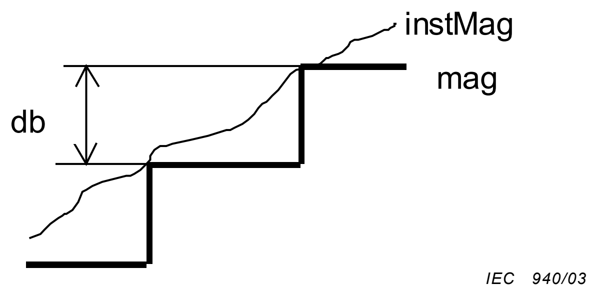


Figure 18 – Deadbanded value (conceptual)

The value of the deadband configuration **db** shall represent the percentage of difference between the maximum and minimum value of the process measurement in units of 0,001 %.

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

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

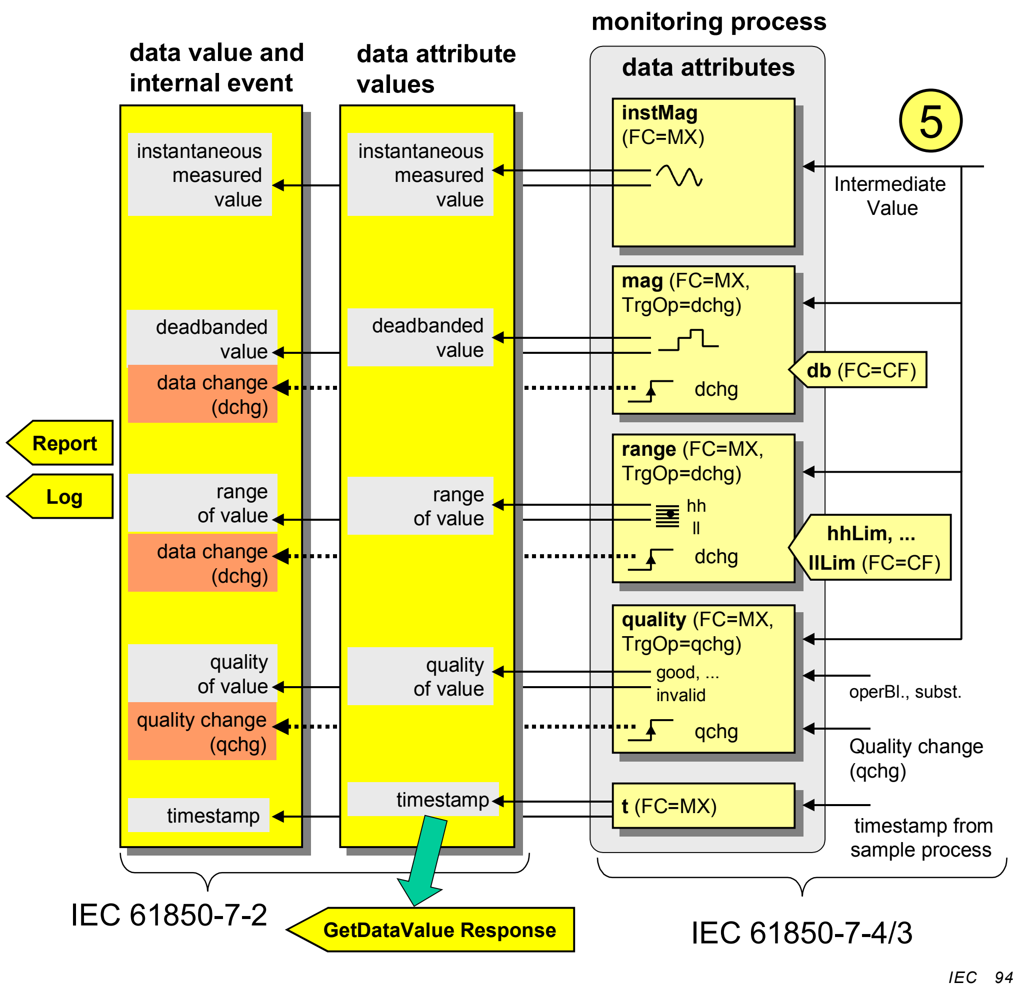


Figure 19 – Input model for analogue values (step 2) (conceptual)

A third application is to monitor the “intermediate value” to determine the current range of the value. The range may be as shown in Figure 20.

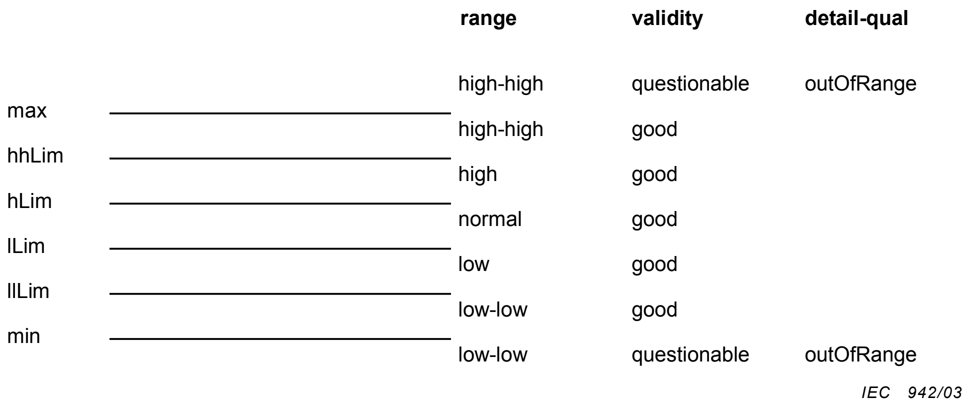


Figure 20 – Range values

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

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

The definitions conceptually depicted on the right hand side of Figure 19 are defined in IEC 61850-7-4 and IEC 61850-7-3. The left hand side and Figure 21 show the definitions (conceptual) found in IEC 61850-7-2.

6.4.3.3 Data reporting and logging

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

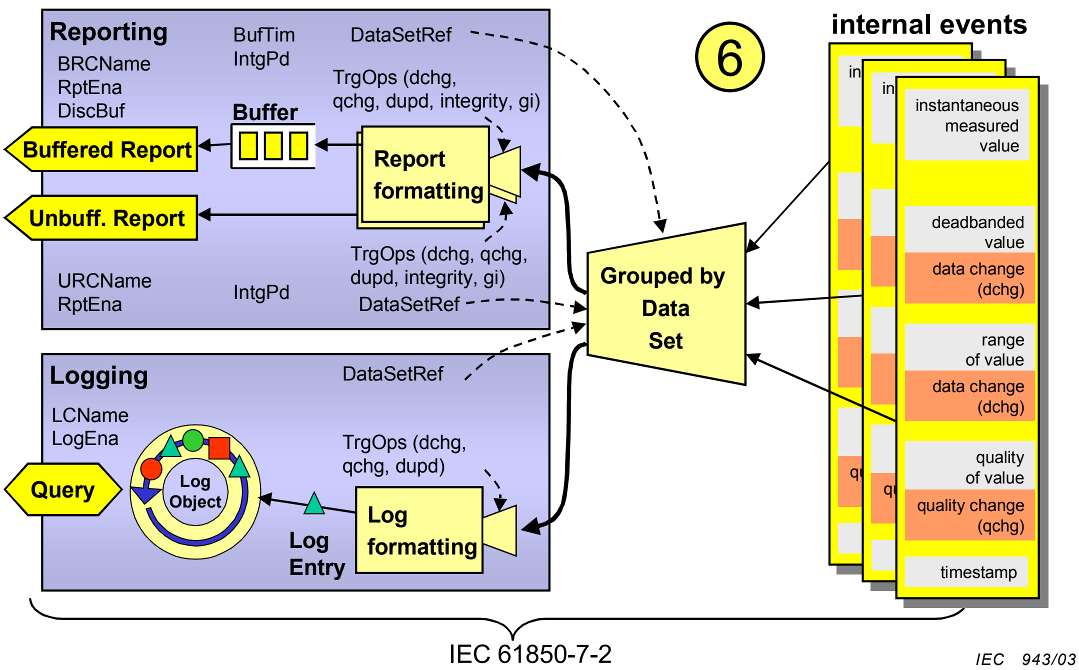
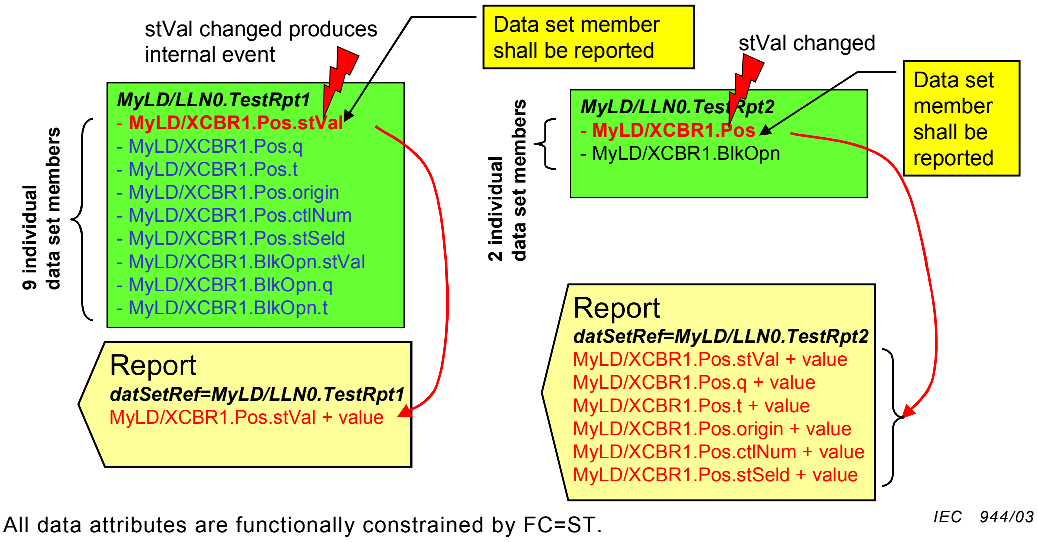


Figure 21 – Reporting and logging model (conceptual)

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

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



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

Figure 22 – Data set members and reporting

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

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

The log model has the log and log control block.

The principle characteristics of the data access methods provided by IEC 61850-7-2 are shown in Table 5.

Table 5 – Comparison of the data access methods

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

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

The basic buffered reporting mechanism is shown in Figure 23. The buffered and unbuffered reporting starts with the configuration of the report control blocks. The reporting starts with setting the enable buffer attribute to TRUE; setting to FALSE stops the reporting.

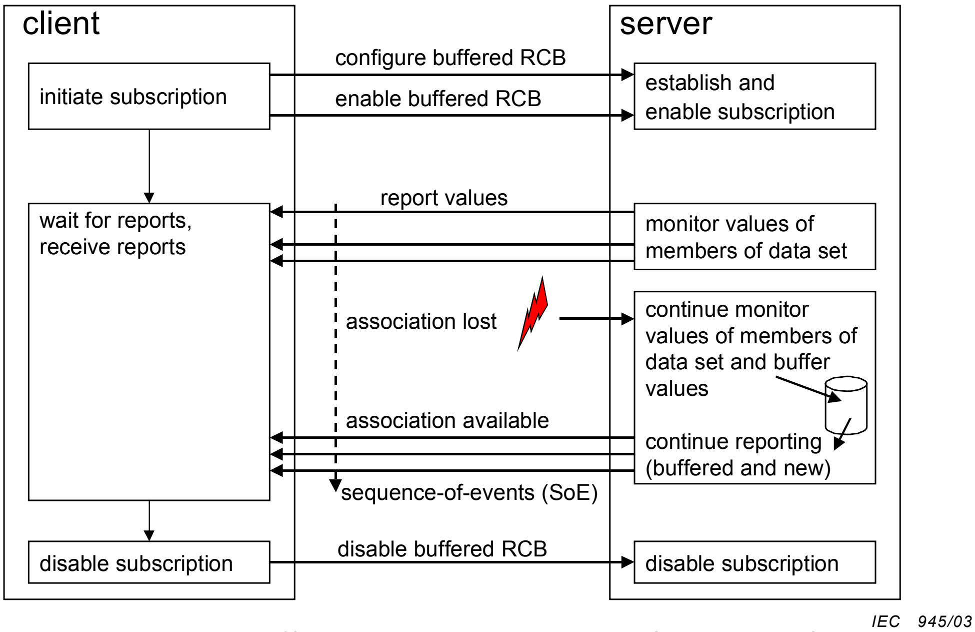


Figure 23 – Buffered report control block (conceptual)

The specific characteristic of the buffered report control block is that it continues buffering the event data as they occur according to the enabled trigger options in case of, for example, a communication loss. The reporting process continues as soon as the communication is available again. The buffered report control block guarantees the sequence-of-events (SoE) up to some practical limits (for example, buffer size and maximum interruption time).

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

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

RpdID handle provided by the client to identify the buffered report control block,

RptEna to remotely enable/disable the reporting process,

DatSet references the data set whose values are to be reported,

ConfRev contains the configuration revision to indicate deletion of a member of the data set or the reordering of the members,

OptFlds indicates the optional fields which are to be included in the report:

– sequence-number to get the correct order of events,
– report-time-stamp to inform the client when the report has been issued,
– reason-for-inclusion to indicate the trigger that has caused the value to be reported,
– data-set-name to indicate from which data sets the values have been generated,
– data-reference to include the object references for the values,

BufTm contains the time to wait after the first event within a data set has occurred (see Figure 24),

SeqNum is the current sequence number of the reports,

TrgOps (trigger options) contains the reasons which causes the control block to report a value into the report. The reasons for reporting may be: data change **dchg**, data update **dupd**, or quality change **qchg** of the data attribute in a logical node,

IntgPd (integrity period): reporting all values initiated by the server based on this period,

GI (general interrogation): reporting all values initiated by the client, and

PurgeBuf set to TRUE indicates to delete all events not yet sent.

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

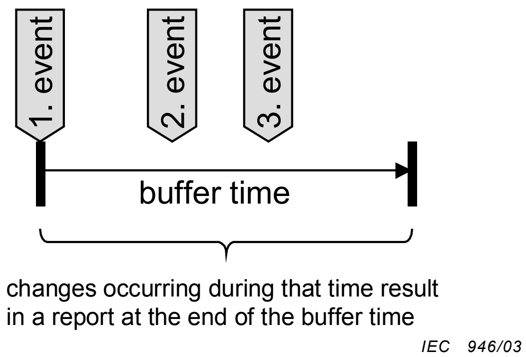


Figure 24 – Buffer time

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

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

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

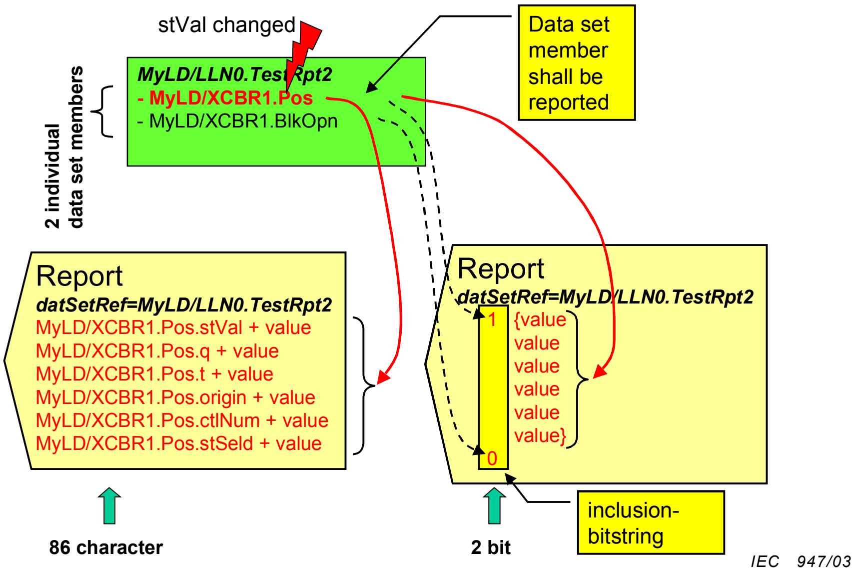


Figure 25 – Data set members and inclusion-bitstring

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

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

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

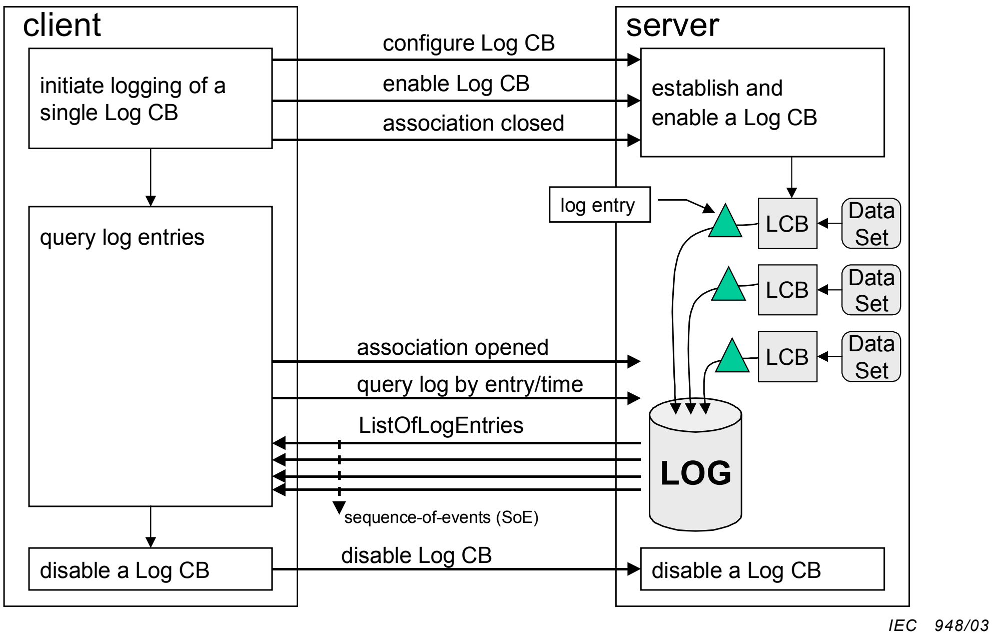


Figure 26 – Log control block (conceptual)

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

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

LogEna to remotely enable/disable the logging process,

DatSet references the data set whose values are to be logged,

TrgOps contains the reasons which cause the control block to store an entry into the log. The reasons for storing a log entry into the log may be: data change **dchg**, data update **dupd**, or quality change **qchg** of the data attribute in a logical node,

integrity period IntgPd: logging all values initiated by the server based on a given period,

LogRef indicating in which log the entries are to be stored.

6.4.3.4 Peer-to-peer data value publishing

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

Figure 27 shows the GOOSE and sampled value models.

NOTE The GSSE (backward compatible with IEEE TR 1550 – UCA – GOOSE) model is similar to the (“IEC”) GOOSE. The GSSE only supports a fixed structure of status data to be published. The data for the GOOSE message is configurable by applying data sets referencing any data.

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

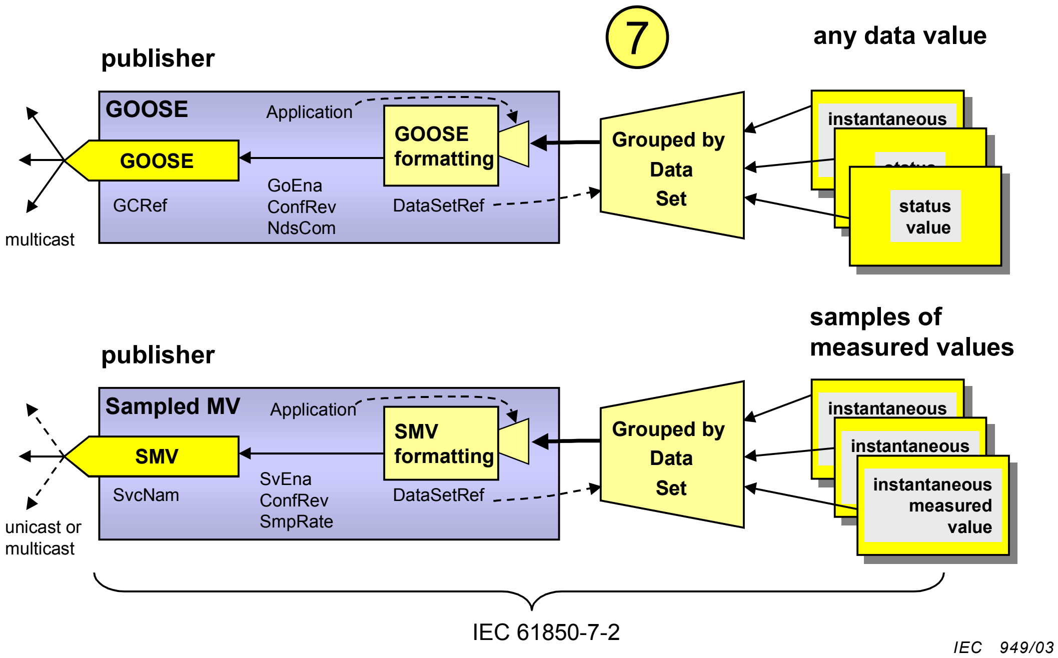


Figure 27 – Peer-to-peer data value publishing model (conceptual)

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

GoEna to remotely enable/disable the publishing,

AppID send in the message to be used as a handle for the receiving application,

DatSet references the data set whose values are to be published,

ConfRev contains the configuration revision to indicate deletion of a member of the data set or the reordering of the members, or changing the DatSet reference,

NdsCom indicates in the message that some commissioning is required.

Which event triggers the publishing of values as well as how often and how fast the values are to be published is outside the scope of this standard.

6.4.3.5 Sampled value publishing

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

SvEna to remotely enable/disable the publishing,

MsvID send in the message to be used as a handle for the receiving application,

DatSet references the data set whose values are to be published,

ConfRev contains the configuration revision to indicate deletion of a member of the data set or the reordering of the members, or changing the DatSet reference,

SmpRate specifies the samples per unit.

7 Application view

7.1 Introduction

A sample operation, illustrated in Figure 28 is to switch a circuit breaker. An operator at a remote HMI wants to remotely switch the circuit breaker. The HMI computer and the circuit breaker have to operate together (interoperate). First, the computer needs to know what information it has to transmit to the IED representing the circuit breaker (normally called the “process interface”). Secondly, it has also to know the name of this IED (for example “Circuitbreaker1”) and how to address the IED. Both the HMI computer on the left side and the IED “circuitbreaker1” on the right side are connected to a common communication network. The HMI sends a control command to the “Circuitbreaker1” to switch the position of the breaker (close the breaker). After switching is completed, the interface IED may (if configured) send a report to the HMI computer indicating that the switch position has changed.

Different users may name the circuit breaker differently: one may use “Circuitbreaker1”, another may chose “CBK-2”. Part 61850-7-4 based on the approach described in IEC 61850-5 standardises many abbreviated names for substation functions and related equipment. The standardised name for a circuit breaker is **XCBR**. This name may be accompanied by a suffix and a prefix: "**Q1XCBR1**" (for naming conventions, see 13.4, A.2, A.3 and IEC 61850-7-2).

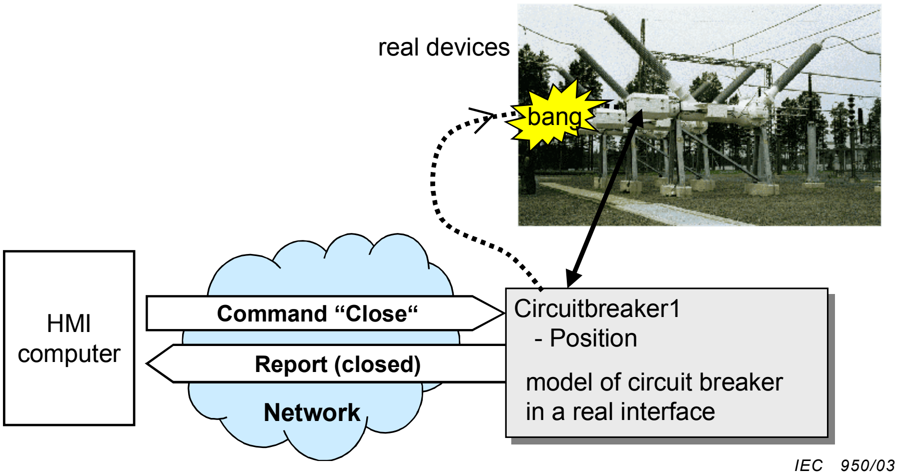


Figure 28 – Real world devices

Applications in the computer on the left hand side of Figure 28 may also query information about the:

– real physical circuit breaker (nameplate, health, ratings, etc.),

– real device (IED) that hosts the process interface (nameplate, health, operation mode, etc.),

– behaviour of the reporting services that determines the transmission of status reports.

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

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

– what functions and what information is network-visible and how are they named and described (IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2),

– how functions can be accessed and how (more generally speaking) information can be exchanged (IEC 61850-7-2), and

– how devices can be connected to communication networks (IEC 61850-8-x and -9-x).

IEC 61850-7-4 comprises a list of more than 2000 named and well defined information elements which enable creation of the information model of a real substation device (for example, complete power transformer, circuit switch, or measurand unit). This static information model inherits the type information from IEC 61850-7-3 and the required (dynamic) communication services from IEC 61850-7-2. Functions in the context of the IEC 61850-7-x series are those that are needed to exchange all information of the information model in the manner that is required for the substation domain. Functions of a substation (for example, bus bar protection, or point of wave switching) make use of the data and functions provided by IEC 61850-7-x.

From the point of view of IEC 61850-7-x, the interactions of logical nodes (other than the services of each logical node and of its data) are outside the scope of IEC 61850-7-x. Examples of interaction of logical nodes for complex functions such as synchronised switching including the basic sequence of exchanged messages can be found in IEC 61850-5.

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

7.2 First modelling step – Logical nodes and data

IEC 61850-7-4 defines a list of some 90 logical nodes. Examples are circuit breaker (abbreviated "**XCBR**") and distance protection ("**PDIS**"). Each logical node (as illustrated in Figure 29) is composed of several data that represent some application specific meaning (see Annex A for an overview of logical nodes and data).

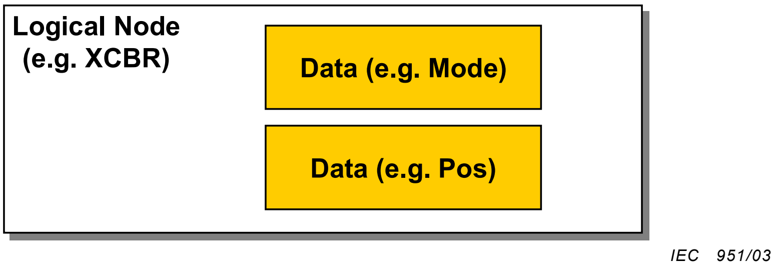


Figure 29 – Logical nodes and data (IEC 61850-7-2)

The data **Pos** as part of a circuit breaker is used to control the position and to report the status of the position; the "**Mode**" represents the current operation mode of the circuit breaker logical node (on, blocked, test, test/blocked, off). This information carries specific meanings in the context of this standard.

As an example, the value “blocked” according to IEC 61850-7-4 means that:

– function of the logical node is active,

– no outputs shall be generated,

– no reports shall be sent,

– control requests shall be rejected,

– all functional and configuration data shall be visible and can be retrieved.

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

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

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

This standard makes use of a hierarchical organisation of data. Figure 30 shows an example of a real device “BayUnit”, protection function functions such as “Distance protection” (**PDIS**) and “Time Overcurrent” (**PTOC**) and “Trip conditioning” (**PTRC**). The process data of these basic functions, the basic functions and other important aspects of the bay unit are modelled as data in a tree-structure. Each element of this tree is a data: the data at the top level is “Bay Unit” which contains “Distance protection”, “Time Overcurrent” and “Trip conditioning”. The “Distance protection” contains, for example, the data “Start” (**Str**) with different attributes such as “general” (**general**) and “Phase A” (**phsA**).

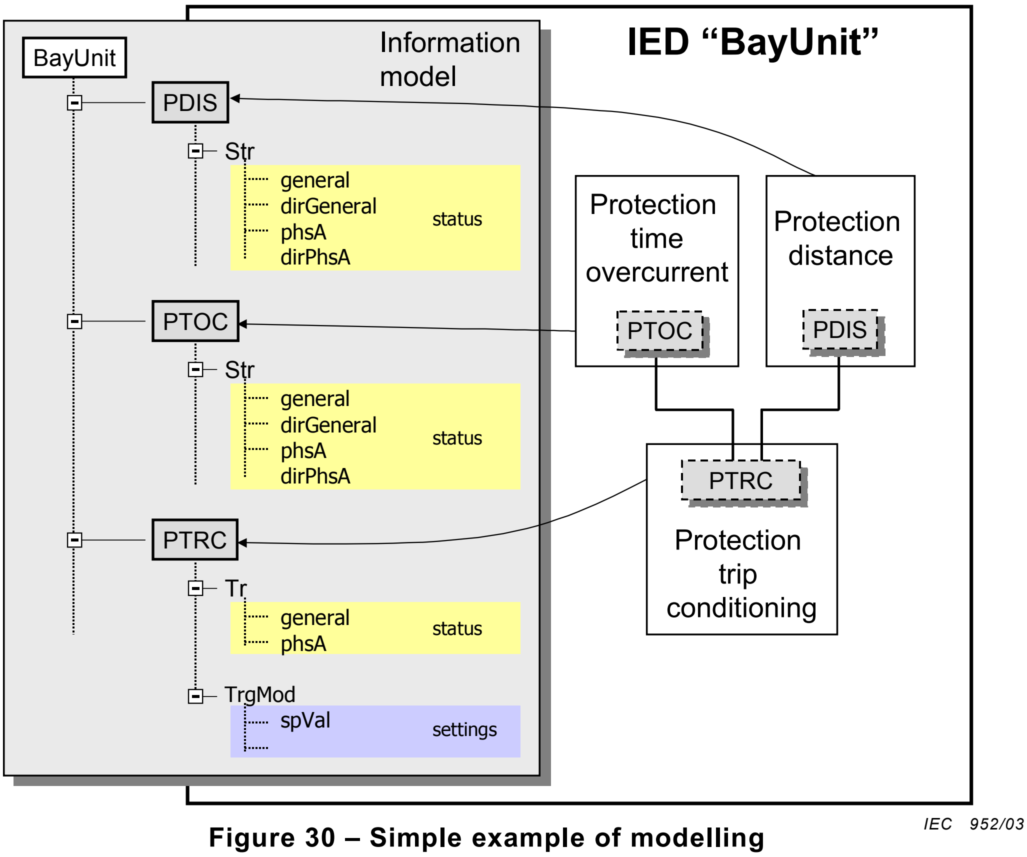


Figure 30 – Simple example of modelling

The elements of a logical node are illustrated in Figure 31. A control service represents the ability to control something in a device. This is modelled as data. To reset, for example, all LEDs in a device, only the value of the data "**LEDRs**" has to be set to True. Data can be grouped into data sets and be reported immediately or logged for later query.

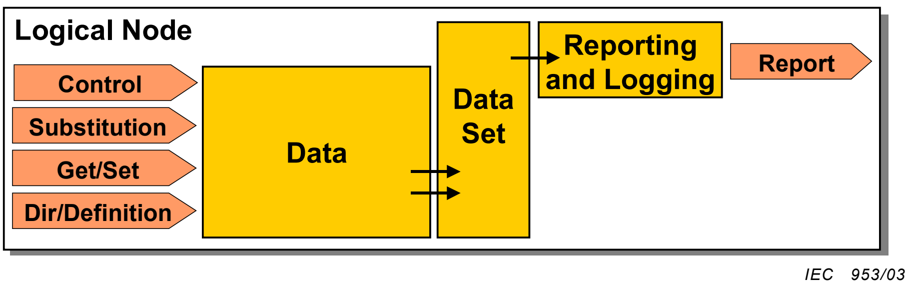


Figure 31 – Basic building blocks

Control and report build one part of the interface of a logical node. Other services that operate on data are: substitution, for replacing a data value with a fixed one; get and set, for reading and writing data and data set values; Dir and Definition (GetDataDirectory, GetDataDefinition) retrieve the directory information of a data instance and the definition of a data instance. From an abstract viewpoint the interfaces of a logical node can be summarised as illustrated in Figure 32. The services can be understood as carrying the information defined by PICOMs (Pieces of communication) as introduced in IEC 61850-5.

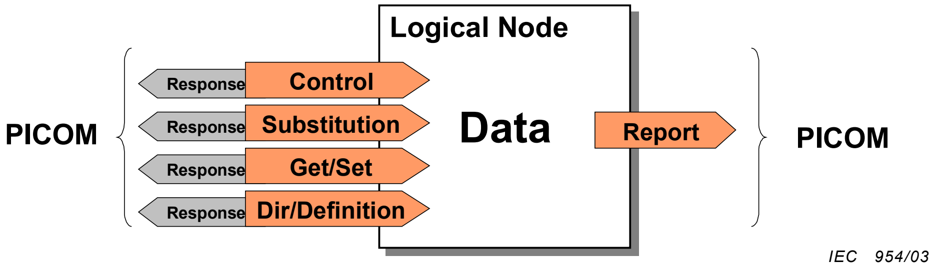


Figure 32 – Logical nodes and PICOM

Logical nodes and data (contained in logical nodes) are the fundamental concepts that are used to describe real systems and their functions. Logical nodes function mostly as a container for data and can be placed anywhere in an IED. Each data defined in IEC 61850-7-4 has a specific meaning assigned to it. The data interact with their environment through their services. The concepts of logical nodes and data in the IEC 61850-7-x series defines the information that can be accessed in a logical node. The device that issues for example the request to retrieve data from a logical node can be modelled as a logical node, too. The information flow can be viewed between logical nodes (see Figure 33 and 9.4).

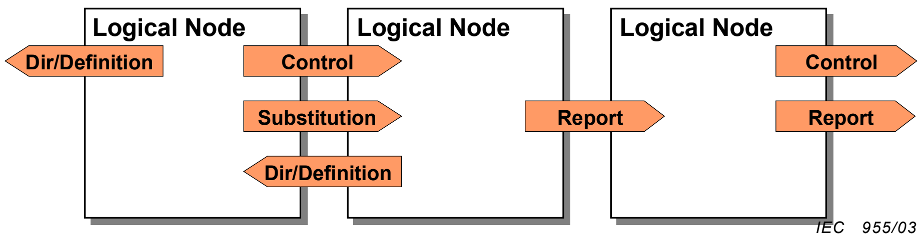


Figure 33 – Logical nodes connected (outside view in IEC 61850-7-x)

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

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

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

8 Device view

8.1 Introduction

Real devices host mainly:

logical nodes and **data** – representing the real application functions and associated information visible from the communication network (data defined in IEC 61850-7-4),

information about real devices – representing information about the resources of the host itself and (if applicable) about the real equipment connected to the host device (specific logical nodes and data defined in IEC 61850-7-4).

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

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

8.2 Second modelling step – logical device model

For communication purposes (beyond a logical node) the concept of a logical device has been introduced. A logical device is mainly a composition of logical nodes and additional services (for example GOOSE, sampled value exchange, setting groups) as illustrated in Figure 34. The grouping of logical nodes in logical devices is based on common features of these logical nodes. For example, the modes of all these nodes are normally switched on and off together, or in the test mode.

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

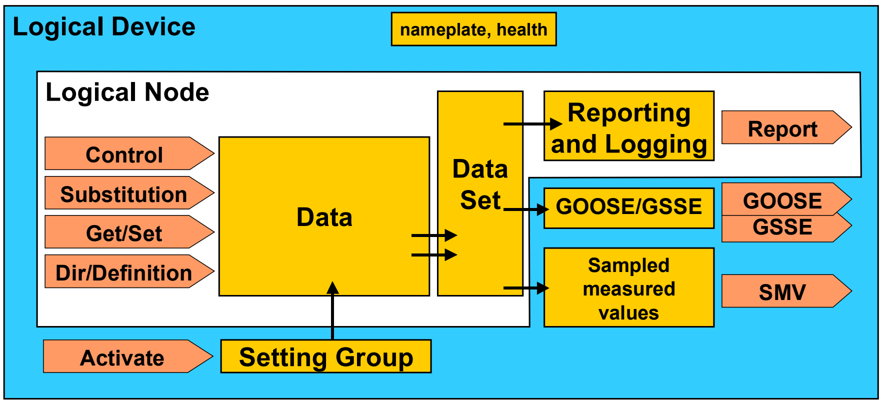


Figure 34 – Logical device building block

In addition logical devices enable the building of gateways (proxies) in such a way that logical devices are – from a functional point of view – transparent. Each logical device can be identified independently of its location (in a separate device connected to the network or in a proxy device).

Logical devices also provide information about the physical devices they use as host (nameplate and health) or about external devices that are controlled by the logical device (external equipment nameplate and health). Logical devices reside in physical devices as shown in the example in Figure 35. Only those aspects of physical devices that are defined as visible to the network are of interest in this standard.

In the example of Figure 35, logical device "LD1" contains three logical nodes. The logical node zero (**LLN0**) represents common data of the logical device, and the logical node physical device (**LPHD**) represents common data of the physical device hosting the logical device. **LLN0** and **LPHD** are defined in any logical device. On the right hand side, the representation of for example nameplate information about the primary equipment is defined as data of the logical node that represents the primary equipment.

LPHD in PHD"A".LD1 provides exactly the same information as **LPHD** in PHD"A".LD2, whereas **LLN0** in PHD"A".LD1 and **LLN0** in PHD"A".LD2 convey different information.

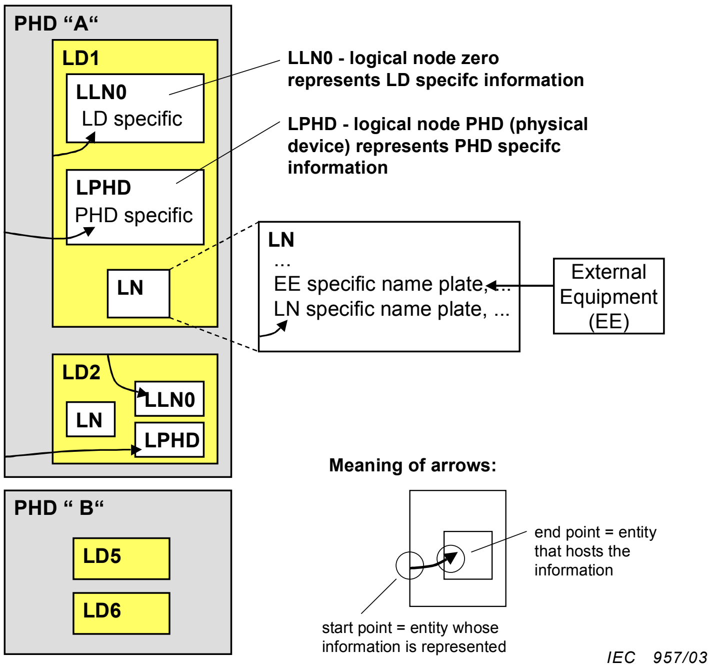


Figure 35 – Logical devices and LLN0/LPHD

Figure 36 shows how multiple physical devices are mapped to a proxy or a gateway. The logical device LD1 is "copied" to the proxy or gateway. The **LPHD** of LD1 in the proxy or gateway represents the physical device **PHD** "A".

To represent the information about the proxy/gateway itself, the logical device **LD0** shall be implemented in each device that acts as a proxy or gateway. The logical nodes **LLN0** and **LPHD** of **LD0** shall represent information about the proxy or gateway device. If a physical device does not provide logical devices that mirror logical devices of other physical devices, then this physical device does not need to provide a **LD0**.

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 (for instance LD7). These logical devices shall have the data **LPHD.Proxy.stVal** of the **LPHD** set to FALSE.

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., LD1). These logical devices shall have the data **LPHD.Proxy.stVal** of the **LPHD** set to TRUE.

LD0 may contain domain specific logical nodes.

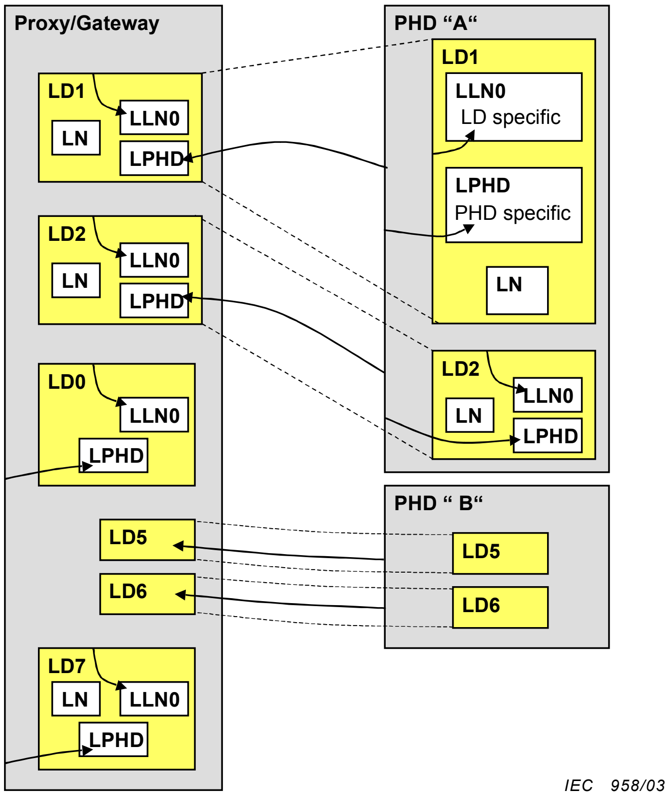


Figure 36 – Logical devices in proxies or gateways

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

9 Communication view

9.1 The service models of the IEC 61850 series

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

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

The ACSI (Abstract communication service interface) defines common utility services for substation devices. The two groups of communication services are depicted in Figure 37. One group uses a client-server model with services like control or get data values. A second group comprises a peer-to-peer model with GSE services (used for time-critical purposes, for example, fast and reliable transmission of data between protection IEDs, from one IED to many remote IEDs) and with the sampled value services for transmissions based on a periodic basis.

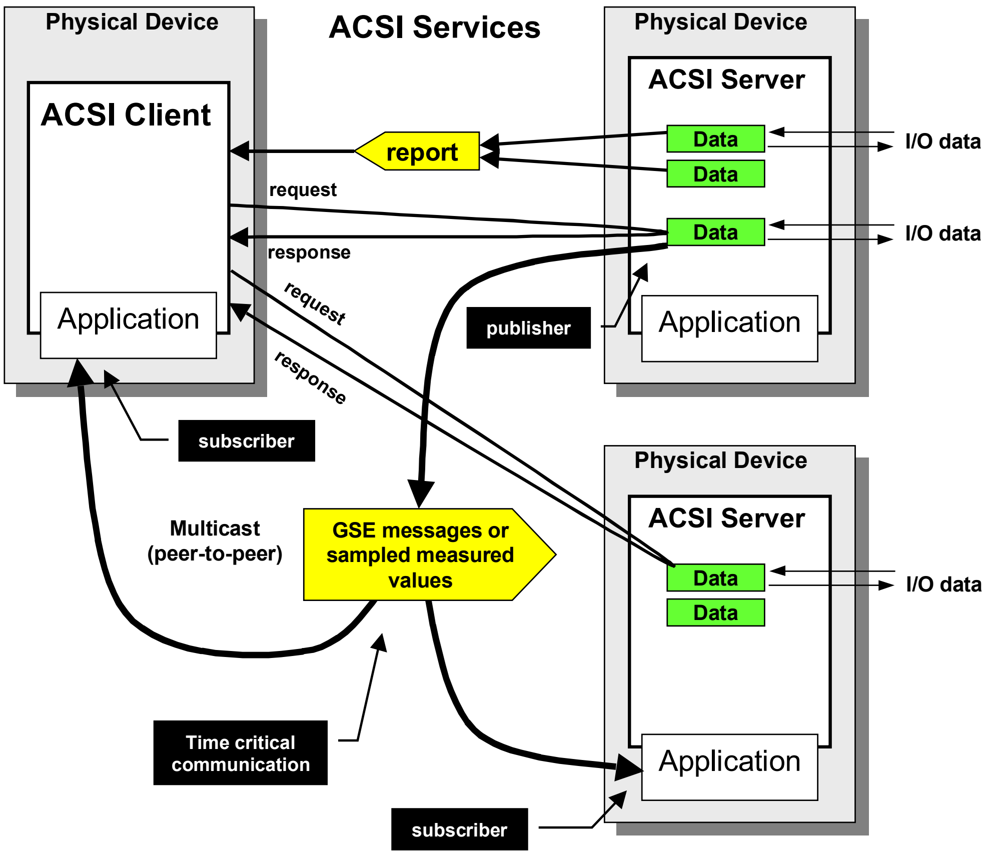


Figure 37 – ACSI communication methods

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

Table 6 lists the ACSI service models and services.

Table 6 – ACSI models and services

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

Service model	Description	Services
Data set	Allow to group various data together.	GetDataSetValue SetDataSetValue CreateDataSet DeleteDataSet GetDataSetDirectory
Substitution	The client can request the server to replace a process value by a value set by the client, for example, in the case of an invalid measurement value.	SetDataValues
Setting group control	Defines how to switch from one set of setting values to another one and how to edit setting groups.	SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues
Reporting and logging	Describes the conditions for generating reports and logs based on parameters set by the client. Reports may be triggered by changes of process data values (for example, state change or deadband) or by quality changes. Logs can be queried for later retrieval. Reports may be sent immediately or deferred (buffered). Reports provide change-of-state and sequence-of-events information exchange.	**Buffered RCB:** Report GetBRCBValues SetBRCBValues **Unbuffered RCB:** Report GetURCBValues SetURCBValues **Log CB:** GetLCBValues SetLCBValues **Log:** QueryLogByTime QueryLogAfter GetLogStatusValues
Generic substation events (GSE)	Provides fast and reliable system-wide distribution of data; peer-to-peer exchange of IED binary status information. GOOSE means Generic Object Oriented Substation Event and supports the exchange of a wide range of possible common data organised by a DATA-SET GSSE means Generic Substation State Event and provides the capability to convey state change information (bit pairs).	**GOOSE CB:** SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues **GSSE CB:** SendGSSEMessage GetGsReference GetGSSEElementNumber GetGsCBValues SetGsCBValues
Transmission of sampled values	Fast and cyclic transfer of samples, for example, of instrument transformers.	**Multicast SVC:** SendMSVMessage GetMSVCBValues SetMSVCBValues Unicast SVC: SendUSVMessage GetUSVCBValues SetUSVCBValues
Control	Describes the services to control, for example, devices or parameter setting groups.	Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate
Time and time synchronisation	Provides the time base for the device and system.	services in SCSM
File transfer	Defines the exchange of huge data blocks such as programs.	GetFile SetFile DeleteFile GetFileAttributeValues

9.2 The virtualisation

The ACSI provides access to the real data and real devices through a virtual image as depicted in Figure 38. A virtual image that represents the real data of devices is made visible and accessible through ACSI services. A computer may request services, for example, get data values, or may receive spontaneously reported values from the controller.

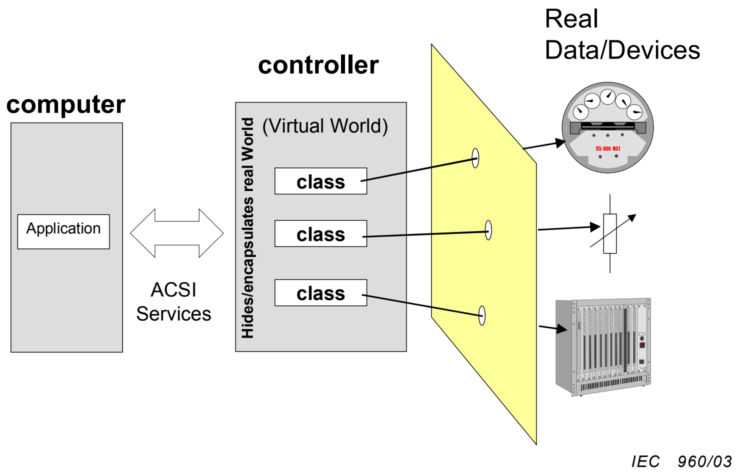


Figure 38 – Virtualisation

The virtual view can be used (as shown in Figure 39) to describe and represent the complete behaviour of a device. Any other device, another controller or even a SCADA system, a maintenance system, or an engineering system may use the ACSI services to inter-operate with that device. A service request received is independent of the device that has requested the service.

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

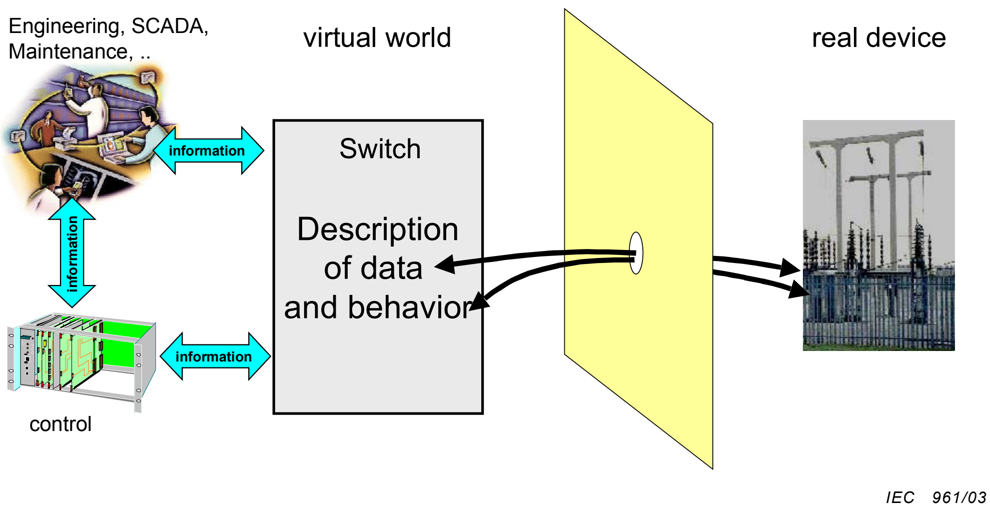


Figure 39 – Virtualisation and usage

9.3 Basic information exchange mechanisms

The ACSI model basically provides the methods of exchanging information between devices as shown in Figure 40.

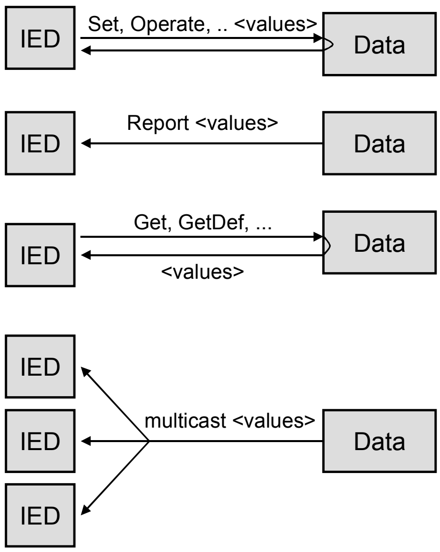


Figure 40 – Information flow and modelling

The use of the generic substation event model (GSE) is quite important because this model supports the implementation of real-time applications. Figure 41 shows an example of an application of the GSE model.

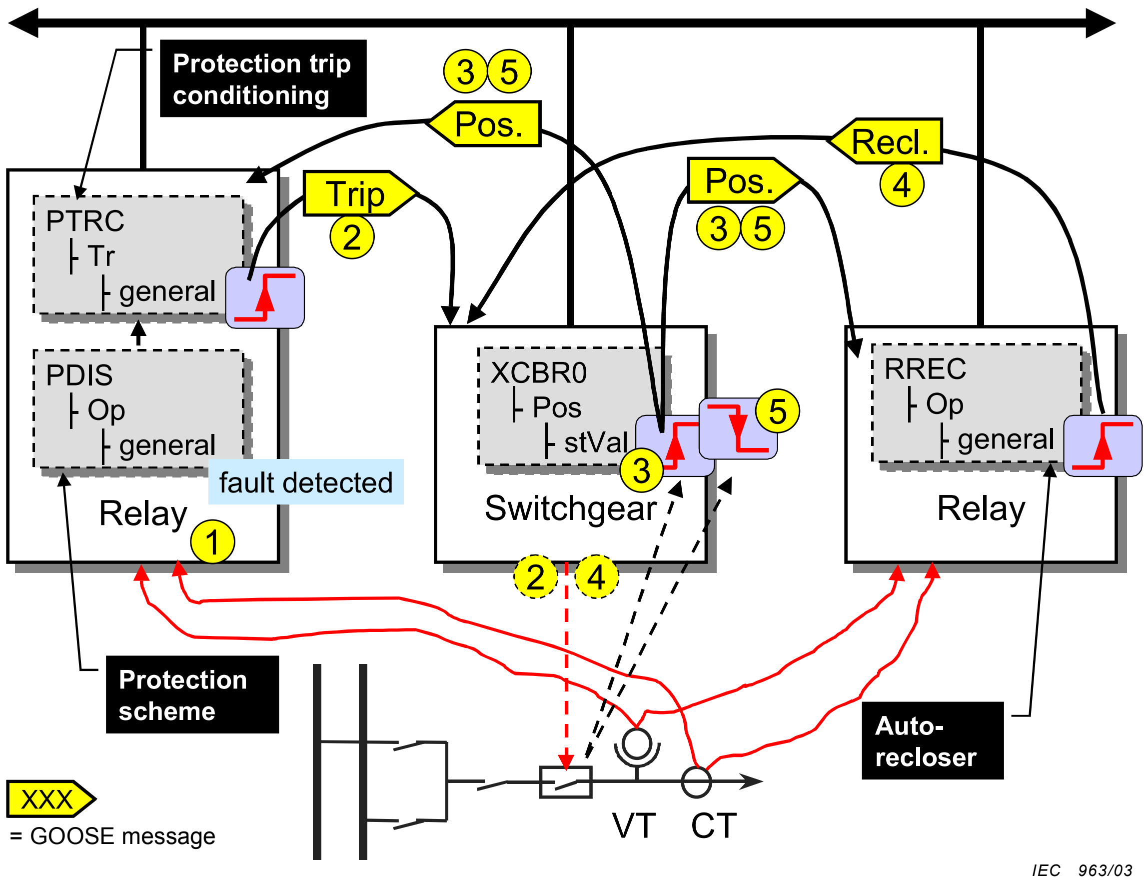


Figure 41 – Application of the GSE model

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

(1) The logical node “protection scheme” (**PDIS**) detects a fault, this results in a decision to issue a trip.

(2) The logical node “protection trip conditioning” (**PTRC**) issues a trip message (applying a GOOSE message), the “circuit breaker zero” (**XCBR0**) has been configured to receive the trip message. After additional processing, the switchgear opens the circuit breaker.

(3) The status information of the “circuit breaker zero” (**XCBR0.Pos.stVal**) changes from ON to OFF. This new state is immediately sent by a GOOSE message with the indication: <new position of switch = open>. In addition, the reporting model may report the change.

(4) The logical node “autoreclosing” (**RREC**) receives a GOOSE message from the **XCBR0** with the value <open>. According to the configured behaviour, the RREC decides to re-close the circuit breaker and sends a GOOSE message with the value <reclose>.

(5) The “circuit breaker zero” (**XCBR0**) receives the GOOSE message with the value <reclose>. After additional processing, the switchgear closes the circuit breaker. The **XCBR** issues sending another GOOSE message < new position of switch = close>.

The sequence is an example only. The IEC 61850 series provides the basic mechanisms for exchanging GOOSE messages under real-time conditions. Applications of GOOSE messaging may be as simple as described in the example. But it may be used in more sophisticated schemes. The trip signal may be repeated once at the very beginning to increase the probability that every receiving IED receives the trip signal. All these schemes are outside the scope of the IEC 61850 series.

9.4 The client-server building blocks

9.4.1 Server

Additional common building blocks provided by the communication system are depicted in Figure 42. The association model provides mechanisms for establishing and maintaining connections between devices and to implement access control mechanisms. Time synchronisation provides the accurate time for time tagging (ms range) in applications such as reporting and logging or for applications such as synchronised sampling (µs range).

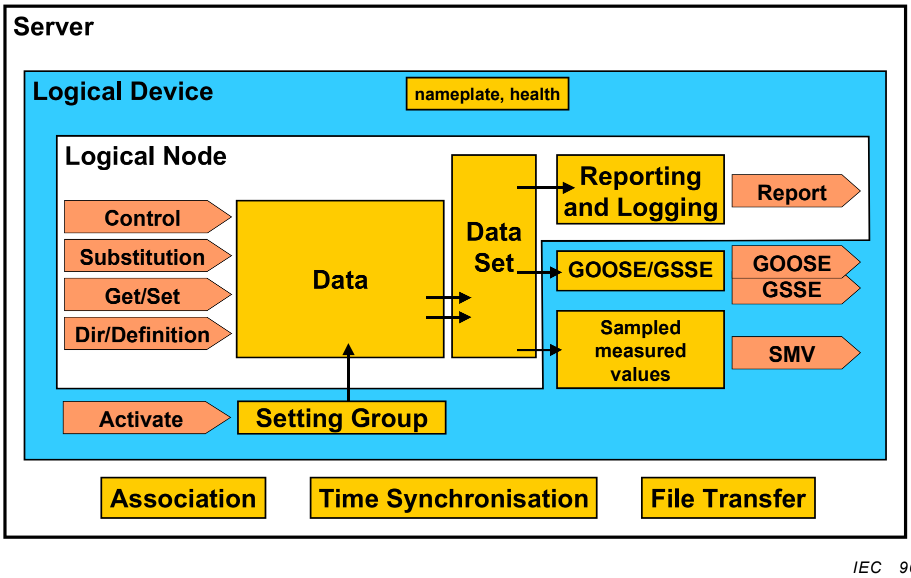


Figure 42 – Server building blocks

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

9.4.2 Client-server

Figure 43 illustrates the client/server role. Clients issue service requests and receive confirmations of the service that has been processed in the server. A client may also receive report indications from a server. All service requests and responses are communicated by the protocol stack that is being used by a specific communication service mapping.

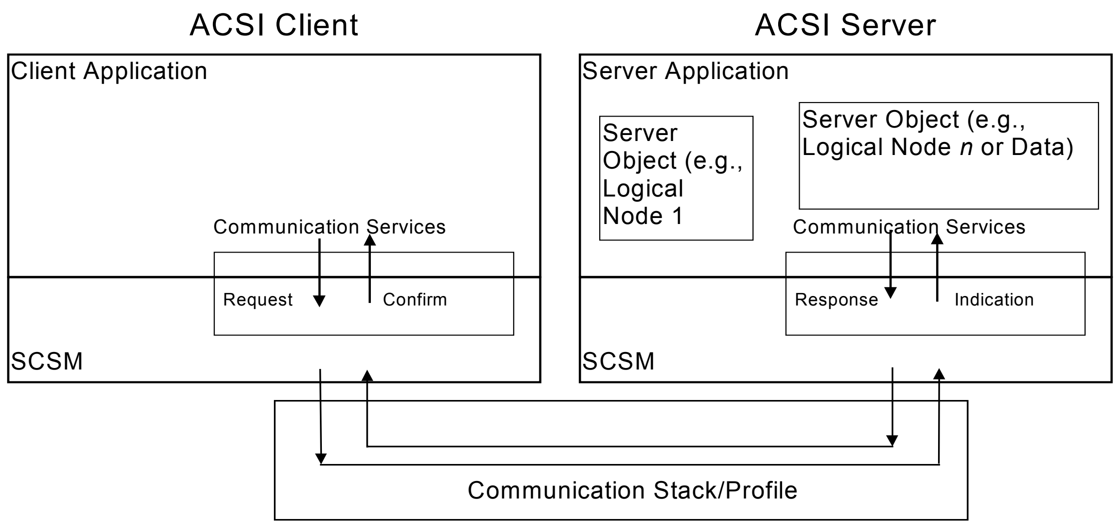


Figure 43 – Interaction between application process and application layer (client/server)

Figure 44 shows an example of a get service that enables a client to retrieve the values of the data inside the server.

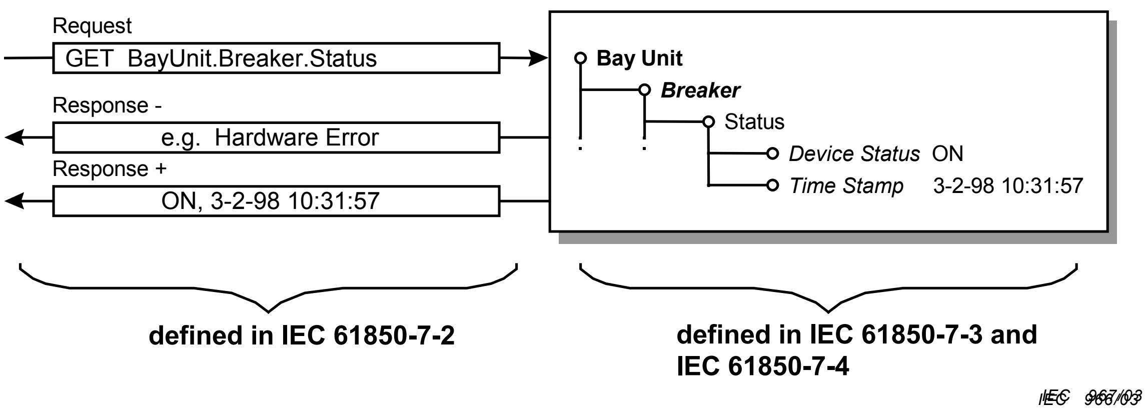


Figure 44 – Example for a service

9.4.3 Client-server roles

According to Figure 45 one server “serves” various logical nodes and clients.

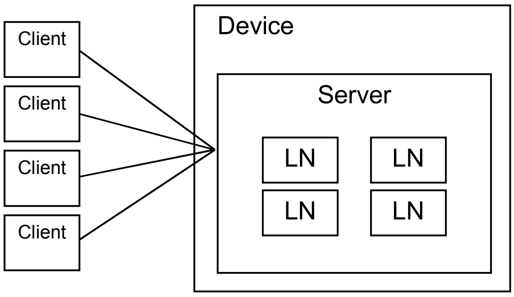


Figure 45 – Client/server and logical nodes

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

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

As shown in Figure 46, the devices may implement the client and the server role.

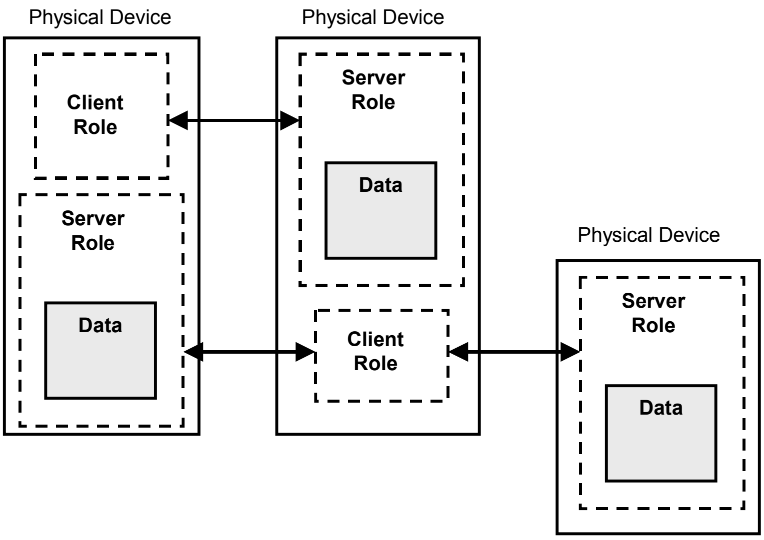


Figure 46 – Client and server role

Logical nodes communicate with other logical nodes by means of PICOMs as described in IEC 61850-5. The logical nodes in that sense comprise the data and control as well as the client and server role (see Figure 47). The client and server are communication-specific entities. From an application viewpoint, they are not required. Therefore, the logical nodes (and only the logical nodes) can be understood as being in communication with one another. This view is just a matter of abstraction.

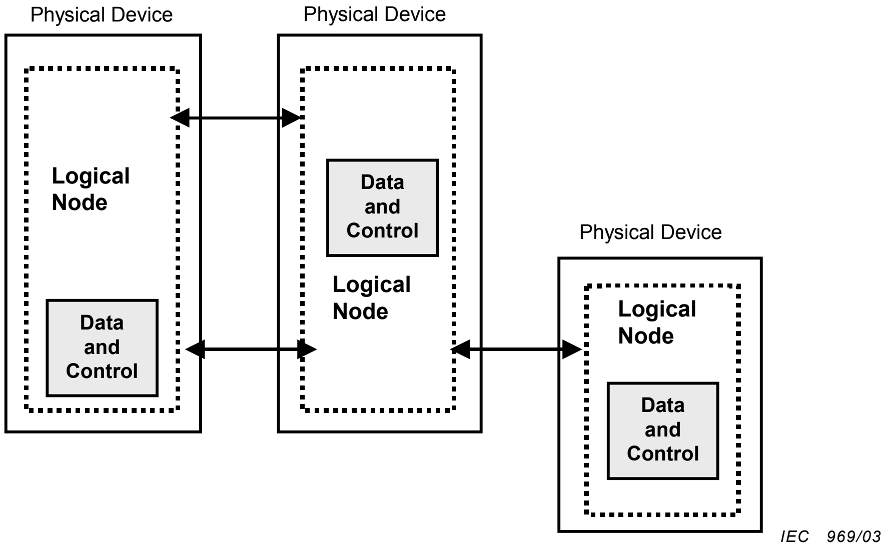


Figure 47 – Logical nodes communicate with logical nodes

The logical node view and the communication view are two different views of the very same real subject.

9.5 Interfaces inside and between devices

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

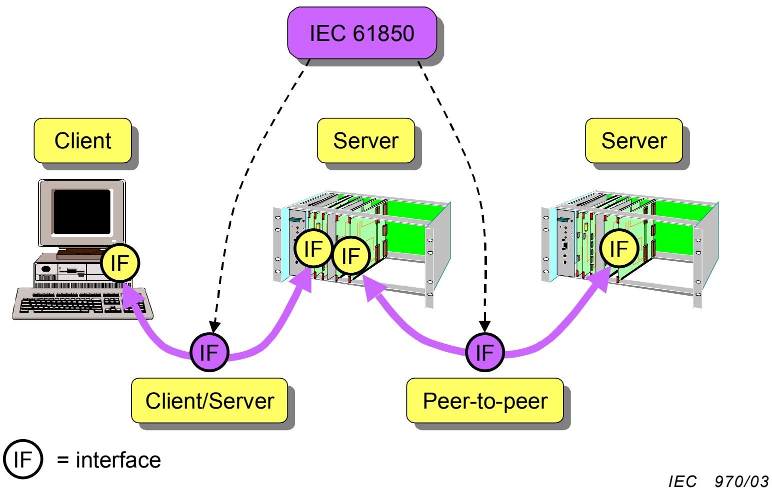


Figure 48 – Interfaces inside and between devices

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

10 Where physical devices, application models and communication meet

Physical devices are placed in the centre of a hierarchy of components as shown in Figure 49. All views “meet” in the server. Each view has a relation to the other views inside the physical device. The various views are shown here to demonstrate that in addition to the IEC 61850 series (which describes just one view of a real automation system) many other aspects have to be taken into account when real devices are implemented.

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

a) the server represents the application data modelling view (the IEC 61850 series) to the outside network,

b) the server represents all aspects of the communication network and the process I/Os to the application of the physical device,

c) a SCSM maps the IEC 61850 series view to the communication network visible objects,

d) the server, the SCSM and the application functional view are mapped to the resource of a physical device.

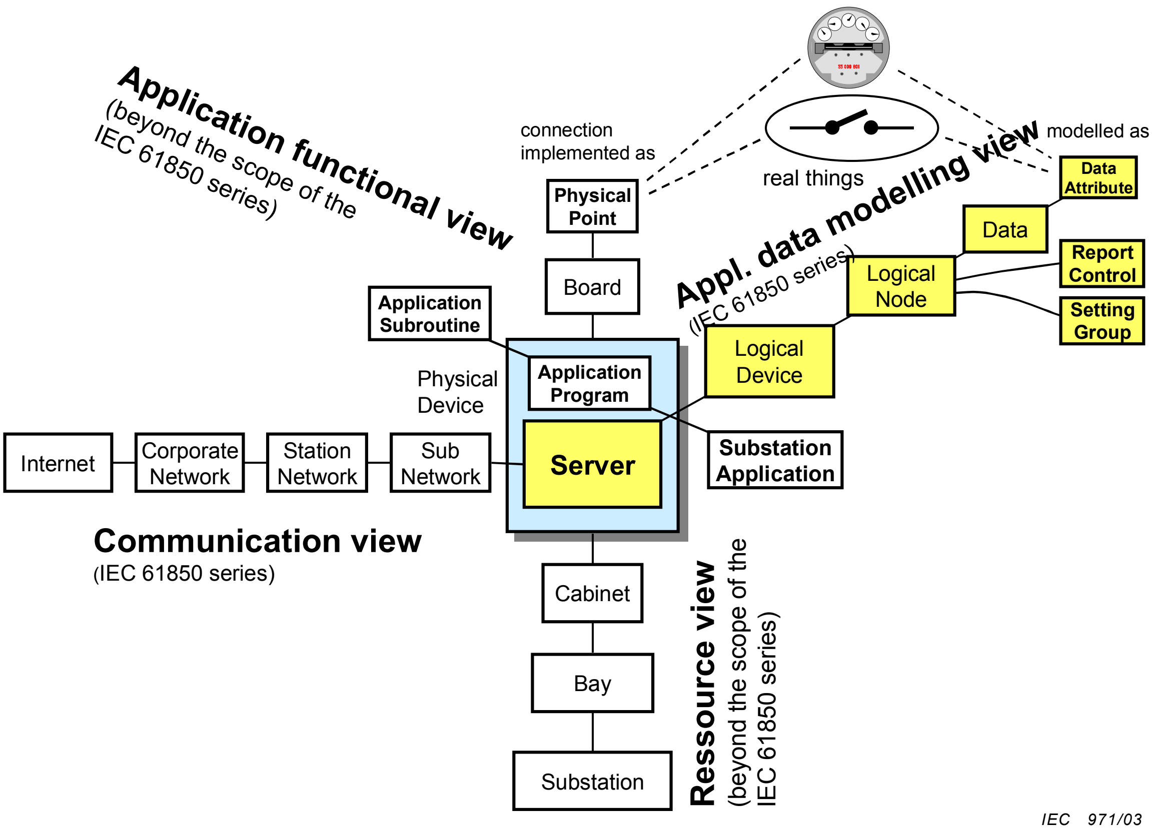


Figure 49 – Component hierarchy of different views (excerpt)

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

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

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

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

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

11.1 Refinements of class definitions

One major building block is the **DATA** class defined in IEC 61850-7-2. The **DATA** class is used in the definition of almost all information that is defined in logical nodes. The **DATA** class as defined in IEC 61850-7-2 is on the left hand side of Figure 50. The **DATA** class defines three data attributes and four services. The services are defined in IEC 61850-7-2. The content of the data attributes are not specified in IEC 61850-7-2. The **DATA** class therefore is very generic. It must become more specific if it is to be used in an application domain. This could require definition of all **DATA** needed to model substation specific functions inside the logical nodes. It is common practice to analyse an application domain to find common properties and terms applicable to many data classes. These common definitions are provided by the common data classes (**CDC**) specified in IEC 61850-7-3.

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

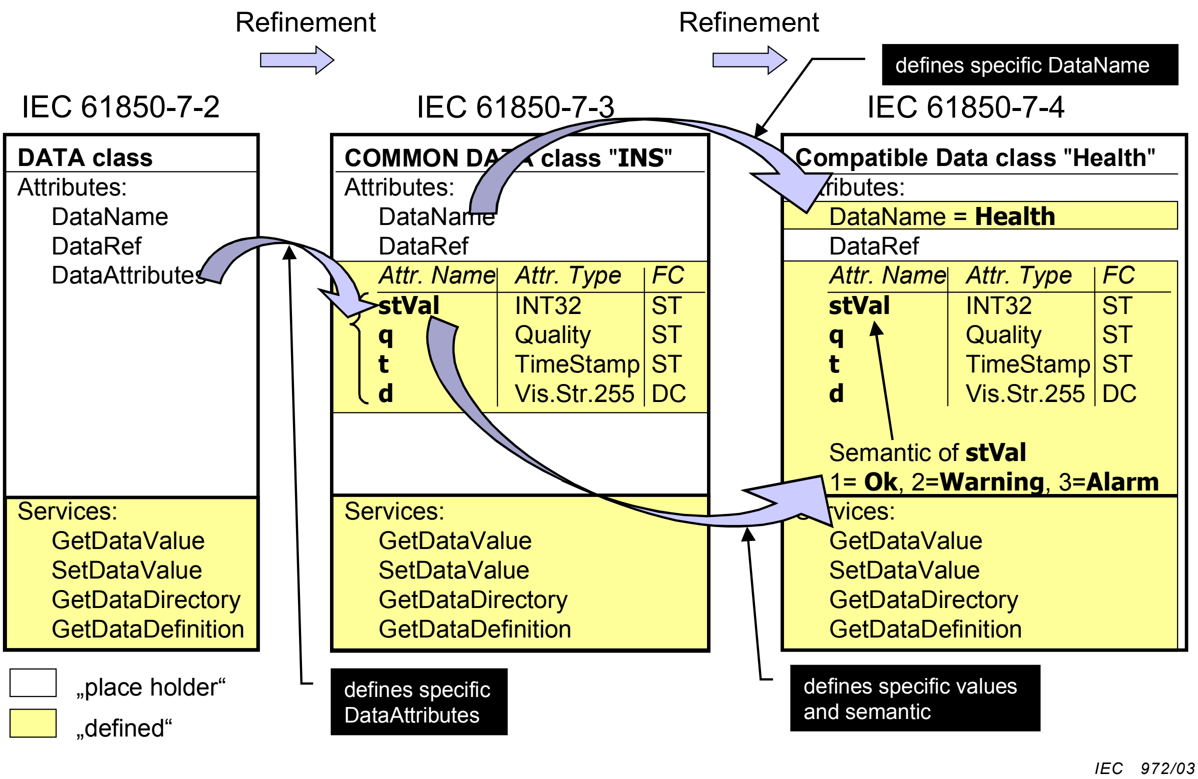


Figure 50 – Refinement of the DATA class

The **DATA** so far does not tell much about its use or the semantics of the data attributes derived from "**INS**". The class on the right hand side of Figure 50 (taken from IEC 61850-7-4) defines exactly this “use”. The "**Health**" class defines the name "**Health**". That name will be used in all instances derived from this class. In addition, the status value "**stVal**" is defined as having three values: “Ok” (=1), “Warning” (=2), and “Alarm” (=3).

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

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

11.2 Example 1 – Logical node and data class

Table 7 shows an example of a list of **DATA** classes for a circuit breaker. The name of the circuit breaker class is "**XCBR**". The **DATA** classes that make up the circuit breaker are grouped into three categories (basic LN information, controllable data, and status information). Each category comprises some **DATA** classes, for example, "**Mode**" and “Switch position”. These **DATA** classes are referenced by their DataName: "**Mode**" and "**Pos**". To be more precise, each **DATA** class also has a common data class, defining the details , i.e. the **ATTRIBUTES** of the **DATA** class. The last column specifies whether this data class is mandatory (M) or optional (O)

Table 7 – Logical node circuit breaker

Logical Node: Circuit breaker		**Name: XCBR**	
Data-Class	**DataName**	**Common Data Class (CDC)**	**M/O**
Basic Logical Node information			
Mode	Mod	INC - Controllable Integer Status	M
Behaviour	Beh	INS - Integer Status	M
Health	Health	INS - Integer Status	M
Name plate	NamPlt	LPL - Logical node name plate	M
Local operation (local means without sub-station automation communication, hard-wired direct control)	Loc	SPS - Single point status	
External equipment health	EEHealth	INS - Integer Status	
External equipment name plate	EEName	DPL - Device name plate	
Operation counter	OpCnt	INS - Integer Status	
Controllable Data			
Switch position	Pos	**DPC - Controllable Double Point**	M
Block opening	BlkOpn	SPC - Controllable Single Point	M
Block closing	BlkCls	SPC - Controllable Single Point	M
Charger motor enabled	ChMotEna	SPC - Controllable Single Point	O
Metered Values			
Sum of Switched Amperes, resetable	SumSwARs	BCR - Binary counter reading	O
Status information			
Circuit breaker operating capability	CBOpCap	INS - Integer Status	M
Point On Wave switching capability	POWCap	INS - Integer Status	O
Circuit breaker operating capability when fully charged	MaxOpCap	INS - Integer Status	O

Since many **DATA** classes use the same details (**ATTRIBUTES**), these details are therefore collected for re-use in common data classes (common to many **DATA** classes). The common data classes are defined in IEC 61850-7-3. As an example, the “controllable double point” (**DPC**) common data class for "**Pos**" is shown in Table 8.

Table 8 – Controllable double point (DPC)

DPC class					
Attribute name	**Attribute type**	**FC**	**TrgOp**	**Value/value range**	**M/O/C**
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
			control and status		
ctlVal	BOOLEAN	CO		off (FALSE) | on (TRUE)	AC_CO_M
operTim	TimeStamp	CO			AC_CO_O
origin	Originator	CO, ST			AC_CO_O
ctlNum	INT8U	CO, ST		0..255	AC_CO_O
stVal	CODED ENUM	ST	dchg	intermediate-state | off | on | bad-state	M
q	Quality	ST	qchg		M
t	TimeStamp	ST			M
stSeld	BOOLEAN	ST	dchg		AC_CO_O
			substitution		
subEna	BOOLEAN	SV			PICS_SUBST
subVal	CODED ENUM	SV		intermediate-state | off | on | bad-state	PICS_SUBST
subQ	Quality	SV			PICS_SUBST
subID	VISIBLE STRING64	SV			PICS_SUBST
			configuration, description and extension		
pulseConfig	PulseConfig	CF			AC_CO_O
ctlModel	CtlModels	CF			M
sboTimeout	INT32U	CF			AC_CO_O
sboClass	SboClasses	CF			AC_CO_O
d	VISIBLE STRING255	DC		Text	O
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
cdcName	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M
Services					
...					

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

At least all the mandatory attributes of all mandatory **DATA** classes of the "**XCBR**" in Table 7 make up the attributes of the "**XCBR**". Optional **DATA** classes (for example, Point On Wave switching capability – **POWCap**) and optional data attributes (for example, **origin** – Originator) shall be used if required by an application.

All (possible) data attributes of the **DATA Pos** derived from the common data class **DPC** are shown on the left hand side of Figure 51. An instance containing all data attributes is depicted in the middle. The **DATA** class **Pos** is contained in the logical device **MyLD** and in the logical node **XCBR1**. The second instance has just the five mandatory data attributes.

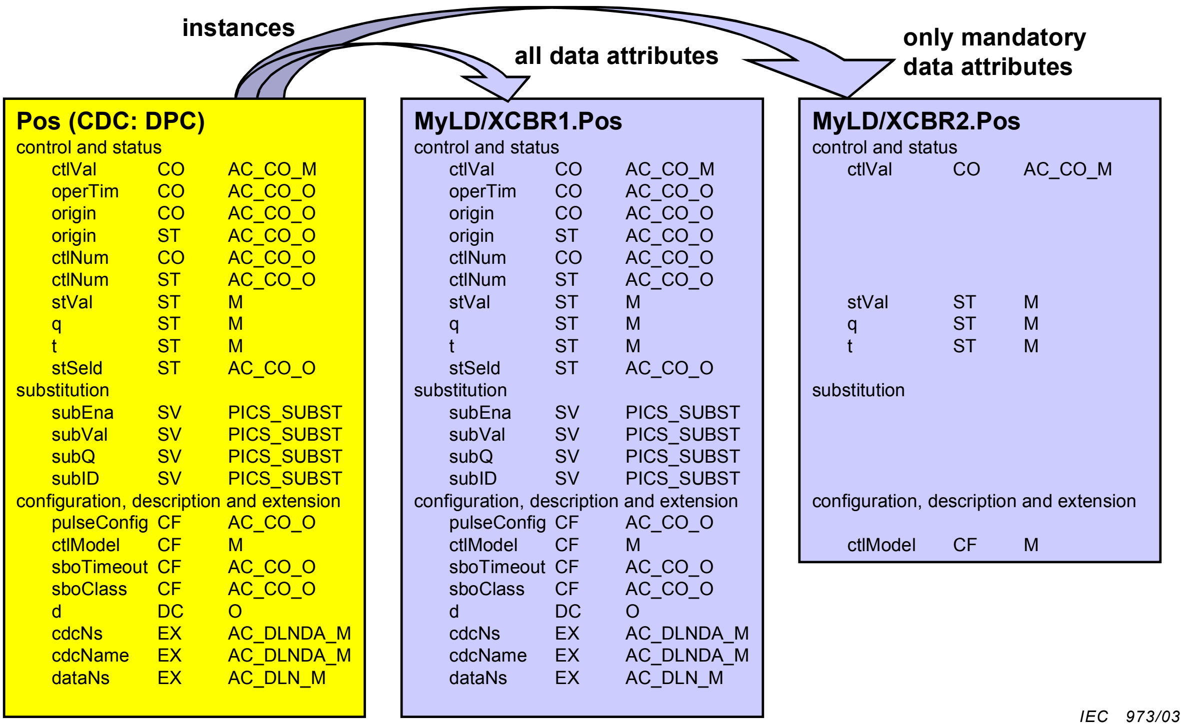


Figure 51 – Instances of a DATA class (conceptual)

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

The conditions in the last column of the **DATA** class **Pos** are as follows (excerpt of IEC 61850-7-3 shown):

Abbreviation	Condition
M	Attribute is mandatory
O	Attribute is optional
PICS_SUBST	Attribute is mandatory, if substitution is supported (for substitution, see IEC 61850-7-2)
AC_DLN_M	Applies to dataNs in all CDCs, dataNs shall be present if the name space of the DATA deviates from the name space defined in ldNs/lnNs
AC_DLNDA_M	The attribute shall be present, if the name space of the CDC deviates from either the name space defined in ldNs/lnNs or the name space defined in dataNs, or both
AC_CO_M	The attribute is mandatory, if the controllable status class supports control
AC_CO_O	The attribute is optional, if the controllable status class supports control
…	…

All 16 **DATA** classes of the circuit breaker class together contain (i.e. when the common data classes are expanded) a total of more than 100 simple data attributes (all mandatory and optional data attributes counted).

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

IEC 61850-7-4 specifies the application-specific semantic for logical node classes and the data classes that belong to logical node classes. The data classes represent structured information, for example, status, quality, or timestamp. A set of common simple and complex structures applicable in most applications are defined in IEC 61850-7-3 (common data classes).

Figure 52 depicts an example of the relation between the three parts. On the level of IEC 61850-7-4, two logical node classes "**XCBR**" and "**XDIS**" are exposed. Each logical node has a data class representing the "controllable double point position" (common data class: "**DPC**"). IEC 61850-7-3 defines a list of some 20 common data classes that can be used to describe the common functionality of data. One logical node instance **XCBR1** is shown at the bottom of Figure 52. This instance can be accessed by services.

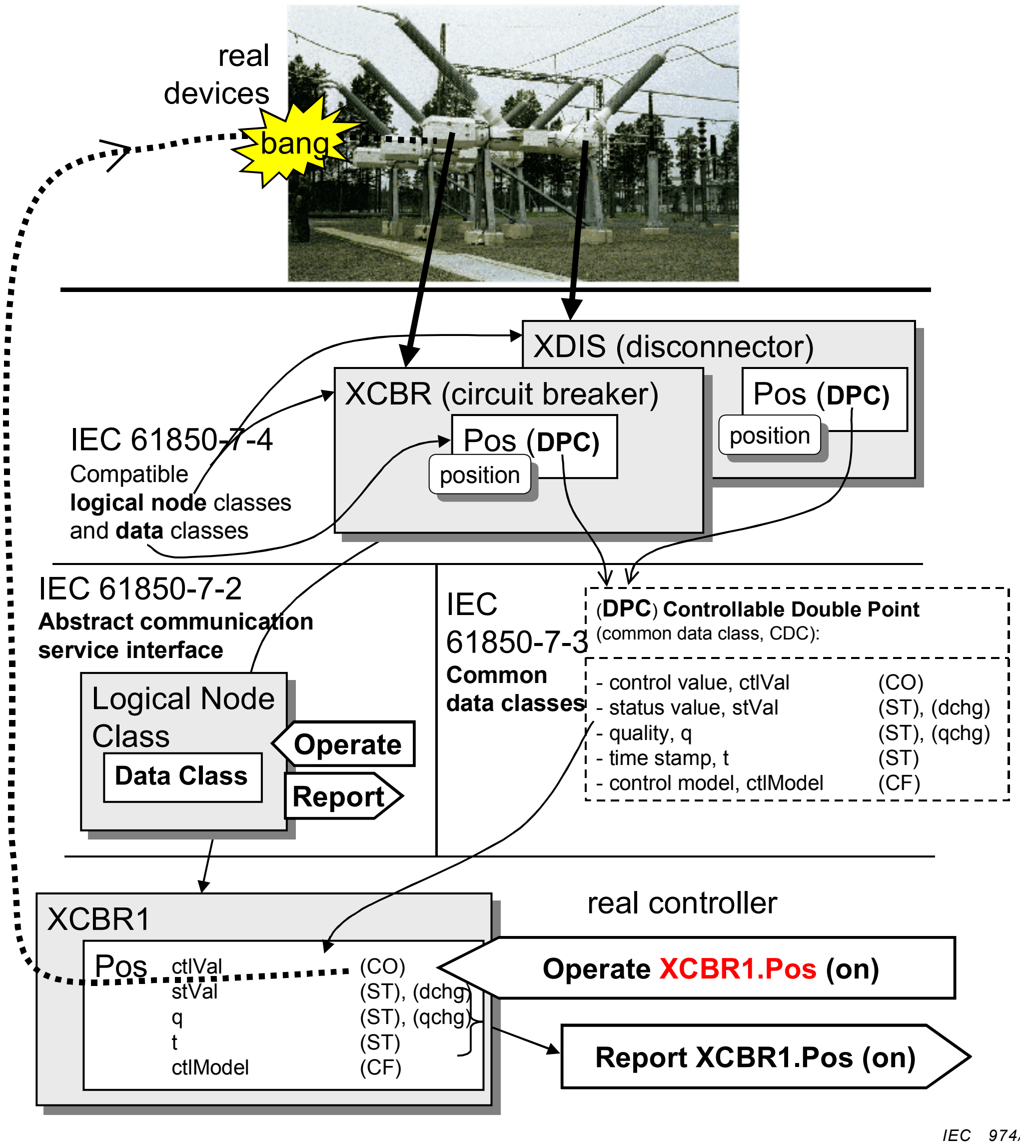


Figure 52 – Relation between parts of the IEC 61850 series

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

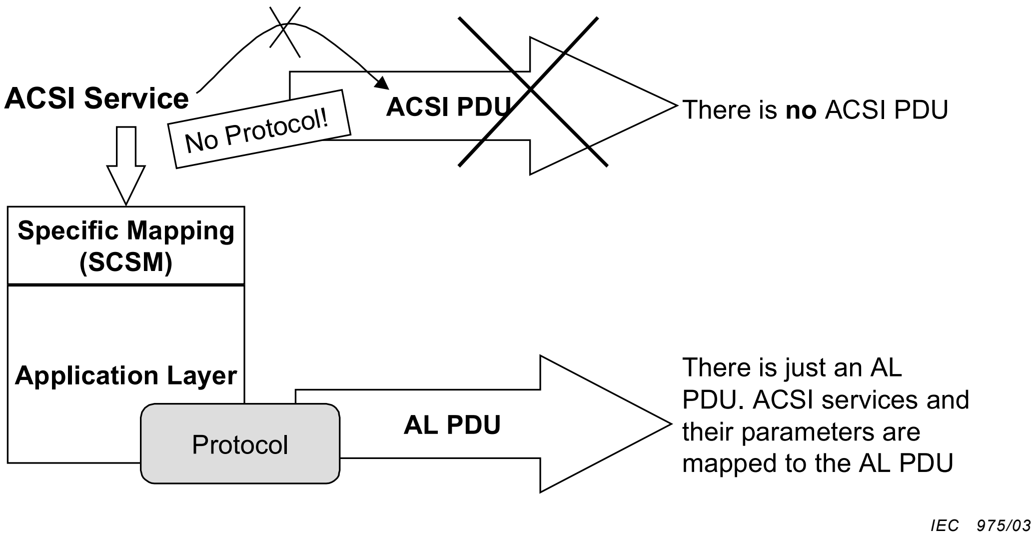
The status value **stVal** has an additional information about when to trigger a report (**dchg** - trigger to send a report on value change of the status value). Reports can also be triggered by changes of the quality **q** attribute **qchg**.

An application of the logical node class and data class (IEC 61850-7-4), common data class (IEC 61850-7-3), and the common logical node, data class, and services in (IEC 61850-7-2) in a real system is shown at the bottom of Figure 52. The service (Operate "**XCBR1**.**Pos**" = on) closes the circuit breaker. The service (Report "**XCBR.Pos**") informs the receiver about the current position change with time stamp and quality information. After a successful switching process, the "**stVal**" data attribute has the new state information.

12 Mapping the ACSI to real communication systems

12.1 Introduction

Figure 53 depicts the relation of the ACSI to an underlying application layer. The ACSI does not define concrete ACSI messages. The ACSI services are mapped to a series of one or more application layer messages (AL PDU – protocol data unit) of the underlying application layer.



PDU: Protocol Data Unit (**encoded** message containing the service parameter, etc.).

Figure 53 – ACSI mapping to an application layer

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

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

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

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

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

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

Standardised mappings of the abstract services to different communication stacks are defined in IEC 61850-8-x and IEC 61850-9-x, so that common utility functions will be performed consistently across all field devices independently of the underlying communication systems. Figure 54 summarises the mappings defined in IEC 61850-8-1, IEC 61850-9-1, and IEC 61850-9-2.

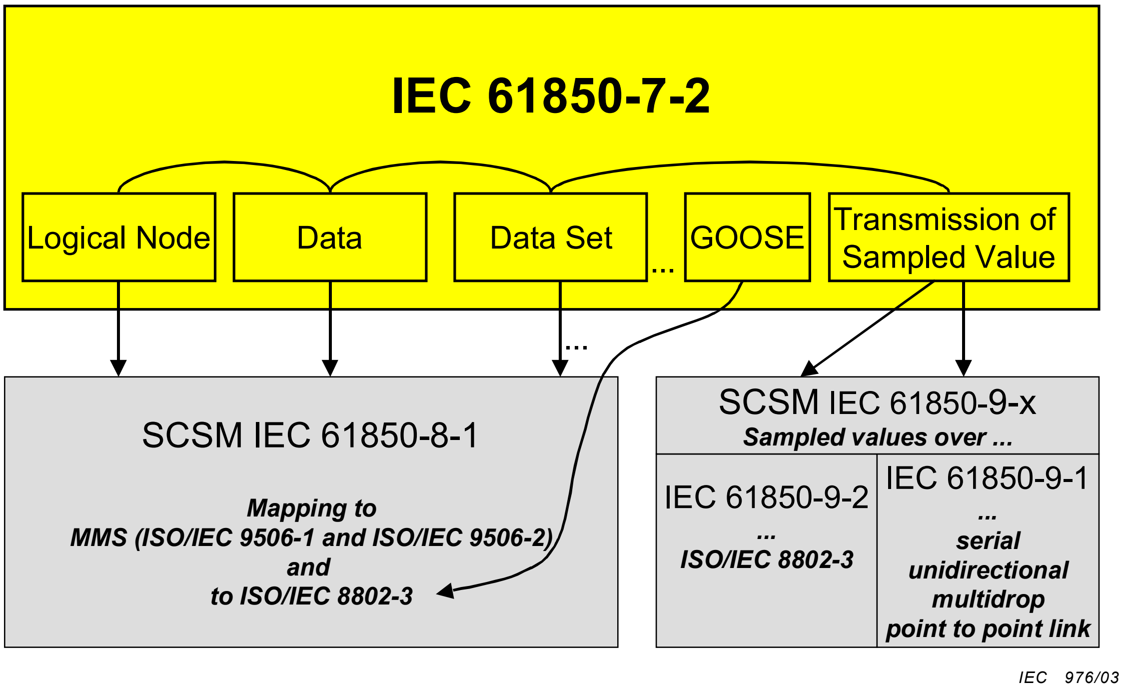


Figure 54 – ACSI mappings (conceptual)

All but GOOSE and transmission of sampled values are mapped to MMS, TCP/IP, and ISO/IEC 8802-3. GOOSE is mapped directly to ISO/IEC 8802-3. The transmission of sampled values is mapped in IEC 61850-9-1 and IEC 61850-9-2.

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

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

According to Figure 55, the SCSM maps the abstract communication services, objects and parameters to the specific application layers. These application layers provide the concrete coding. Depending on the technology of the communication network, these mappings may have different complexities, and some ACSI services may not be supported directly in all mappings, but equivalent services shall be provided (see example below). An application layer may use one or more stacks (layer 1 to 6).

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

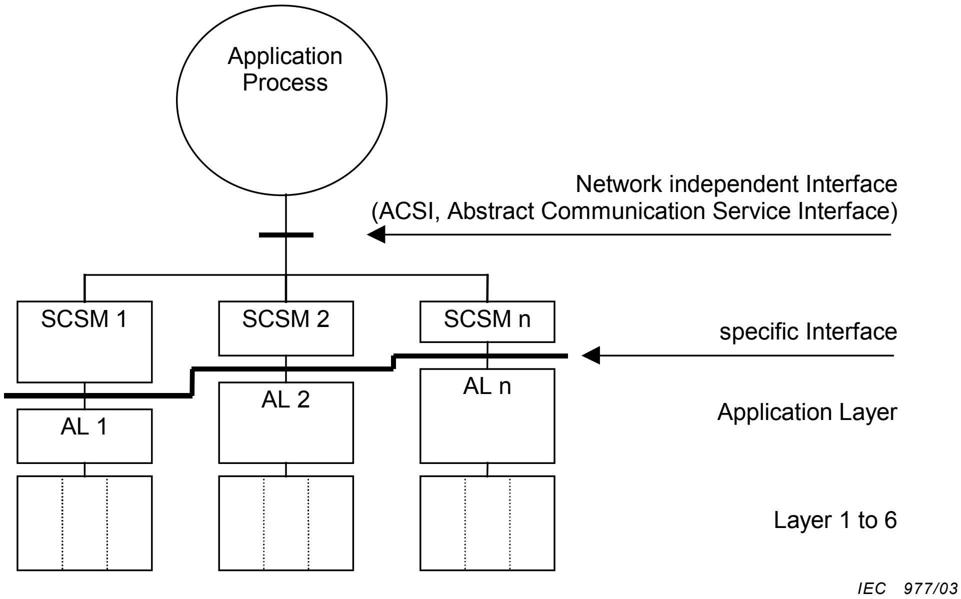


Figure 55 – ACSI mapping to communication stacks/profiles

12.2 Mapping example (IEC 61850-8-1)

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

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

The abstract models of the three parts IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2 need to be mapped to a application layer (see Figure 56).

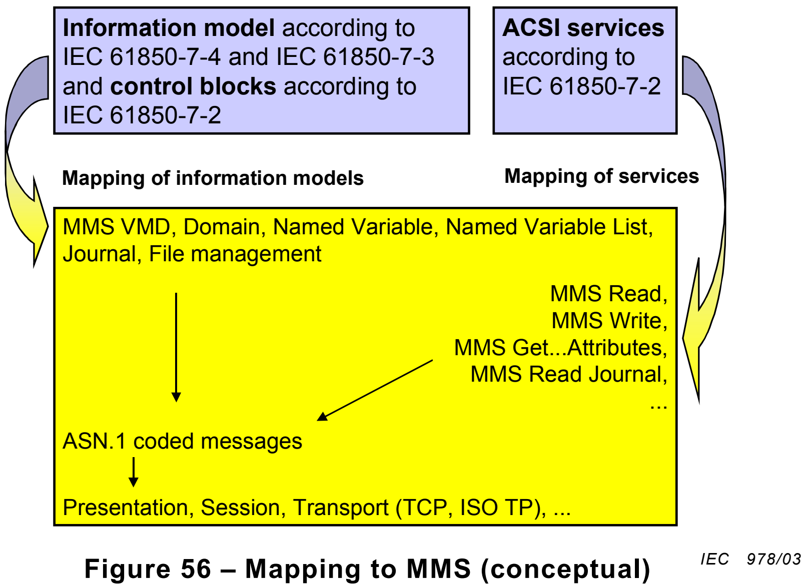


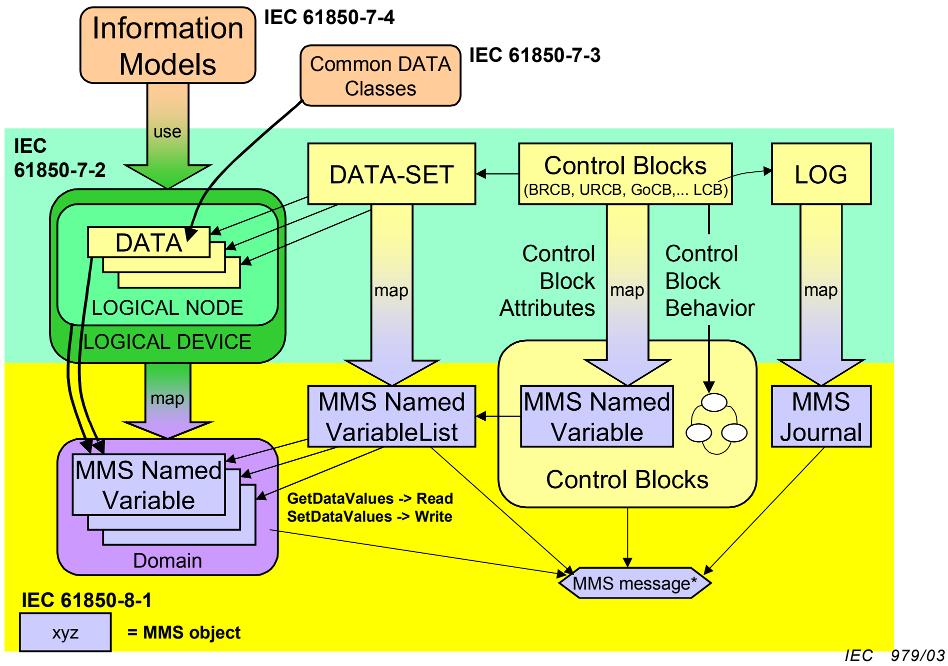
Figure 56 – Mapping to MMS (conceptual)

The information model and the various control blocks are mapped in this example to the Manufacturing Message Specification (MMS), i.e., to the Virtual Manufacturing Device (VMD), domain, named variable, named variable list, journal, and file management. The services are mapped to the corresponding services of the MMS classes.

A more detailed mapping is shown in Figure 57.The abstract information models defined in IEC 61850-7-4, IEC 61850-7-3 and IEC 61850-7-2 map as follows:

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

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



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

Figure 57 – Mapping approach

The details of the mapping to the MMS named variable are sketched in Figure 58.

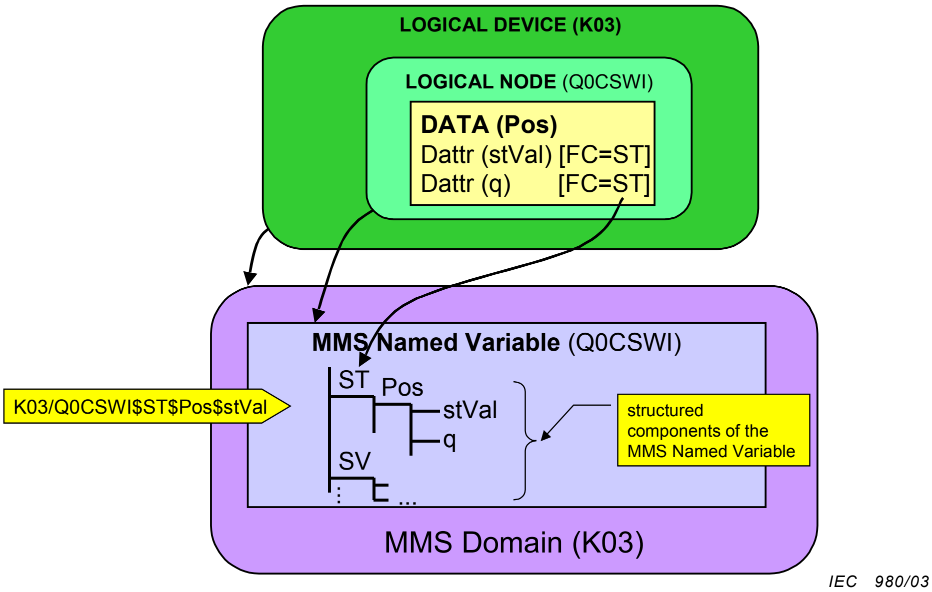


Figure 58 – Mapping detail of mapping to a MMS named variable

The MMS Domain (with the name **K03**) contains named Variables. The named variable shown in Figure 59 has the name "**Q0CSWI**". The components of this named variable are constructed by selecting all data attributes with the same functional constraint (FC), for example, the value FC=**ST** (all status data attributes). The first component of the named variable has the component name "**ST**". The **DATA** (for example, **Pos**) are placed at the next nesting level. The data attributes (for example, **stVal**, **q**, **t**, etc.) are at the next level below. The dots "." in the hierarchical name have been replaced by "$" in the MMS mapping.

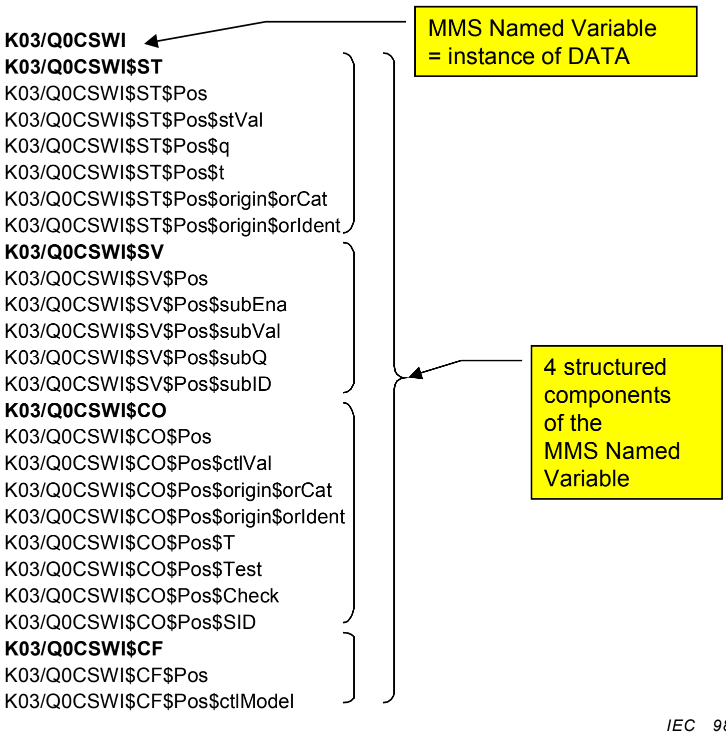


Figure 59 – Example of MMS named variable (process values)

The process values (from the position) of several switch controllers **Q0CSWI$Pos**, **Q1CSWI$Pos**, and **Q2CSWI$Pos** are mapped to MMS named variables (see right lower corner of Figure 60). The position information is grouped by the named variable List (data set) with the name **K03/LLN0$AllRpts**. The attributes of the unbuffered report control block are mapped to the MMS named variable **K03/LLN0RPAllRpts**.The components of this named variable can be written (configured). The control block references the data set.

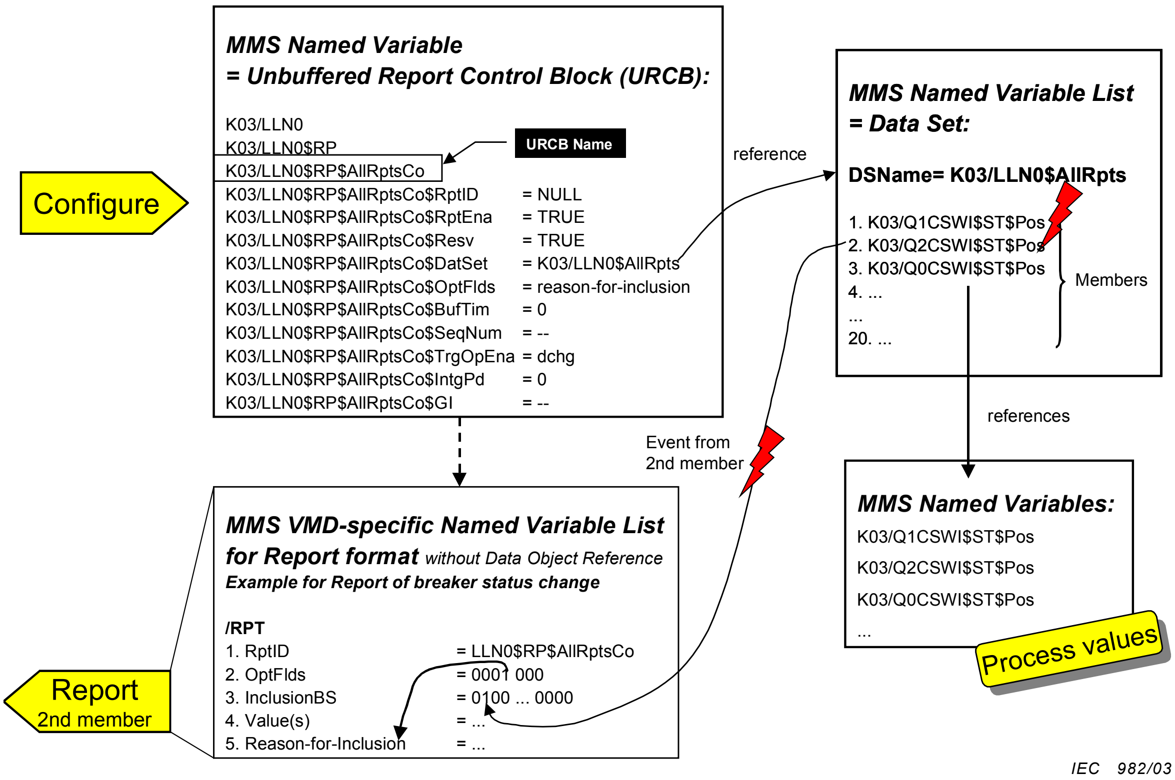


Figure 60 – Use of MMS named variables and named variable list

A change in one of the members of the data set (for example, in member 2) issues sending a report with the status of the position of Q2CSWI. The report message is generated using another MMS named variable list (left lower corner). The report will be sent immediately.

The report is mapped to an MMS information report (see Figure 61). The figure shows the concrete encoding according to ASN.1 BER (the basic encoding rule for abstract syntax notation number one – ISO 8825).

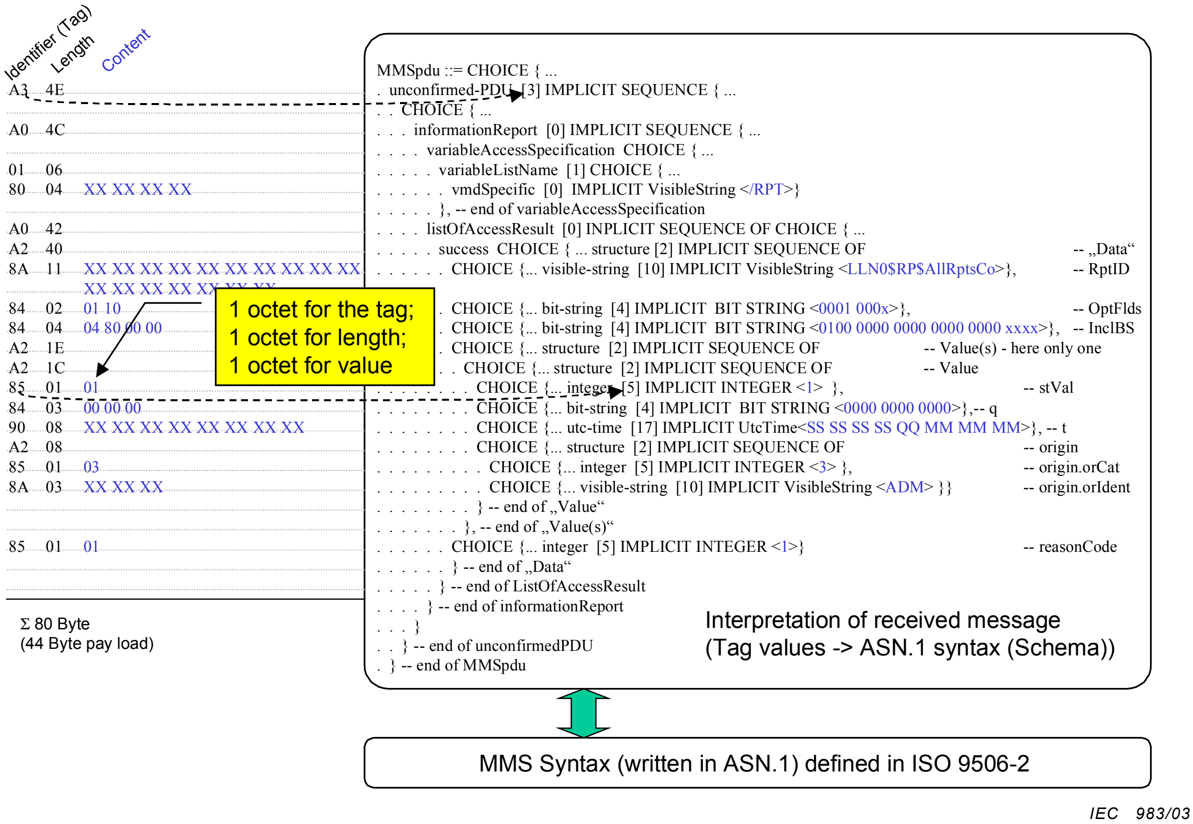


Figure 61 – MMS Information Report message

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

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

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

Figure 62 illustrates a model excerpt of **XCBR1** representing a real device. The complete hierarchical model may be mapped, for example, to MMS applying the SCSM according to IEC 61850-8-1. As a result, many MMS named variables have to be implemented in a real server. The services of the ACSI are mapped to MMS services.

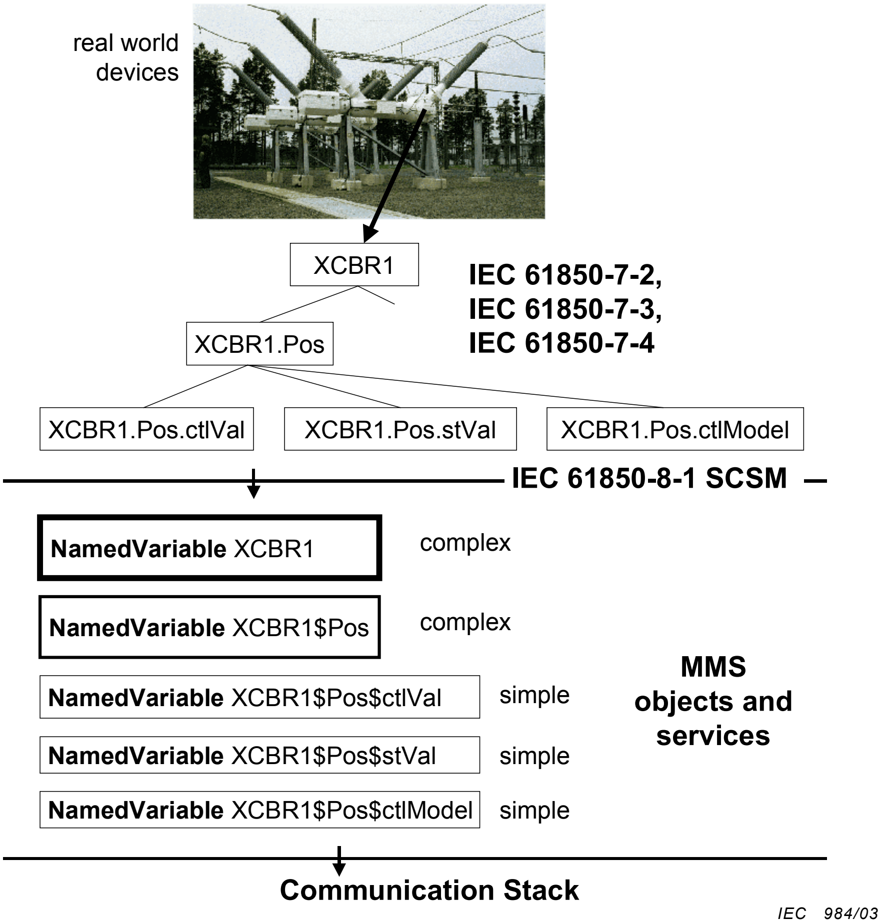


Figure 62 – Mapping example

This example shows that the named variable **XCBR1** represents the logical node (including all **DATA** as components of this named variable). Each component (for example, Pos) has been mapped to a less complex named variable, for example, with the name (with components **ctlVal**, **stVal**, and **ctlModel**). These components are mapped to three even less complex named variables: **XCBR1PosctlVal**, **XCBR1PosstVal**, and **XCBR1PosctlModel**.

NOTE 2 This multi-mapping does not require multiple storages of one value (for example, for **ctlVal**). The tree with all components and sub-(sub-)components is implemented once. The named variable **XCBR1PosctlVal** is just the “address” of the leaf in this tree.

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

13 Formal specification method

13.1 Notation of ACSI classes

IEC 61850-7-2 shall use the class notation as depicted in Table 9.

Table 9 – ACSI class definition

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

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

[0..n] – attribute may be available zero to *n* times

[1..n] – attribute may be available one to *n* times

[0..1] – attribute may be available zero or one times

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

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

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

13.2 Class modelling

13.2.1 Overview

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

Figure 63 depicts the class model of IEC 61850-7-x. The notation used has been derived from UML. The main elements used are the "aggregation" (black diamond) indicating that, for example, the server class is composed of (1 to *n*) logical device classes. Logical device classes comprise many logical node classes (*). Logical nodes are specialisations of the left hand side logical node class (depicted as an open arrow), for example **PDIS** or **XCBR** as defined in IEC 61850-7-4. Logical node classes comprise many data classes. Data classes are specialisations of **DATA** Class, for example, **MDD** – specialised SPS class; or **POS** – specialised DPC class. Finally, common data classes, for example, **DPC**, are composed of (1 to *n*) data attributes.

Example instances are shown on the right hand side, for example, **myXCBR1**:**XCBR** means that **myXCBR1** is an instance derived from the logical node class **XCBR**.

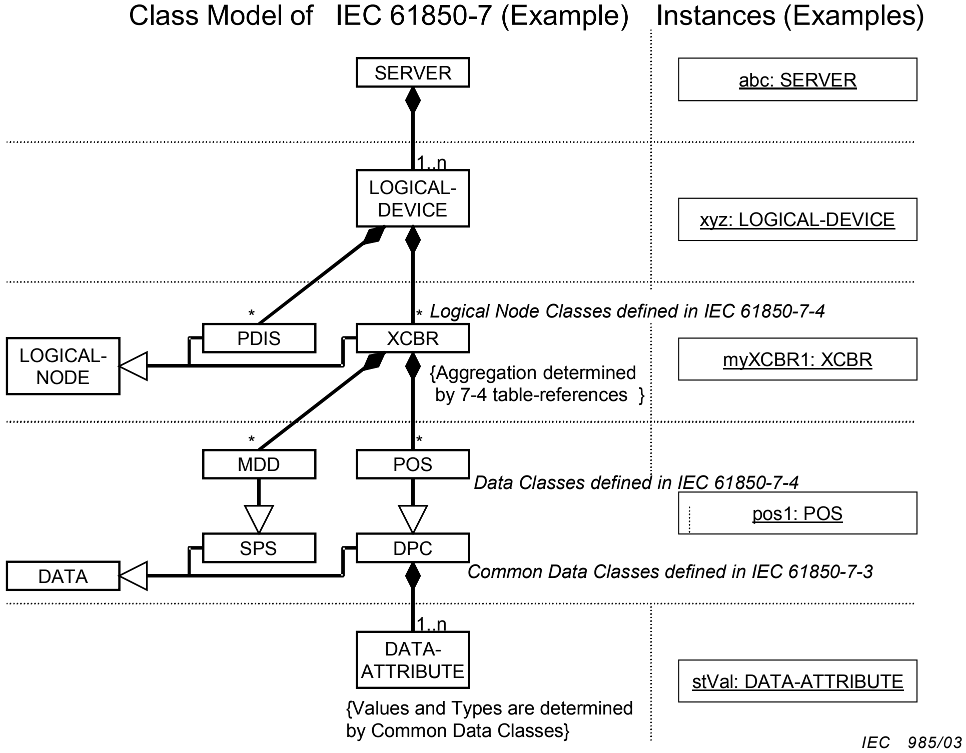


Figure 63 – Abstract data model example for IEC 61850-7

The example instances of Figure 63 demonstrate the instance names: the server is named "abc" of class server etc. The name of the status value "**stVal**" is: "**xyz/myXCBR1.pos1.stVal**"

Each class is characterised by a number of attributes that describe the externally visible feature(s) of all instances of this class. Each instance of a class uses the same attribute types, but has specific values (the instance-specific values) for the attributes. The values of these attributes are defined by IEC 61850-7-x or may be established by IEC 61850-7-x services; hence a change in the device may be modelled by a change in one or more attribute values of an instance.

The following Subclauses discuss examples of the structure of a classes defined in the IEC 61850 series.

13.2.2 Common data class

The attributes of the single point status class are (according to IEC 61850-7-3) defined as depicted in Table 10.

Table 10 – Single point status common data class (SPS)

SPS class					
Attribute name	**Attribute type**	**FC**	**TrgOp**	**Value/value range**	**M/O/C**
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
				status	
stVal	BOOLEAN	ST	dchg	TRUE | FALSE	M
q	Quality	ST	qchg		M
t	TimeStamp	ST			M
				substitution	
subEna	BOOLEAN	SV			PICS_SUBST
subVal	BOOLEAN	SV		TRUE | FALSE	PICS_SUBST
subQ	Quality	SV			PICS_SUBST
subID	VISIBLE STRING64	SV			PICS_SUBST
				configuration, description and extension	
d	VISIBLE STRING255	DC		Text	O
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
cdcName	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M
Services					
As defined in Table 12.					

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 11 (excerpt of **Quality** type of IEC 61850-7-3).

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.

Table 11 – Quality components attribute definition

Quality type definition			
Attribute name	**Attribute type**	**Value/value range**	**M/O/C**
	PACKED LIST		
validity	CODED ENUM	good | invalid | reserved | questionable	M
detailQual	PACKED LIST		M
overflow	BOOLEAN		M
outOfRange	BOOLEAN		M
badReference	BOOLEAN		M
oscillatory	BOOLEAN		M
failure	BOOLEAN		M
oldData	BOOLEAN		M
inconsistent	BOOLEAN		M
inaccurate	BOOLEAN		M
source	CODED ENUM	process | substituted DEFAULT process	M
test	BOOLEAN	DEFAULT FALSE	M
operatorBlocked	BOOLEAN	DEFAULT FALSE	M

The FC column specifies the functional constraint, if applicable. The functional constraint indicates which services can be used to access the values of the data attributes. Table 12 shows which services are allowed for the status information.

Table 12 – Basic status information template (excerpt)

Basic status information template			
Services (see IEC 61850-7-2)			
The following services are inherited from IEC 61850-7-2. They are specialised by restricting the service to attributes with a functional constraint as specified below.			
Service model of IEC 61850-7-2	**Service**	**Service applies to Attr with FC**	**Remark**
Data model	SetDataValues GetDataValues GetDataDefinition	DC, CF, SV **ALL** **ALL**	
Data set model	GetDataSetValues SetDataSetValues	**ALL** DC, CF, SV	
Reporting model	Report	**ALL**	As specified within the data set that is used to define the report content

The services applicable are listed in the third column. For all data that inherit the attributes from the common data class SPS (see Table 12) the attributes with FC=**ST** can be accessed by the following services (indicated by the key word **ALL**):

GetDataValues

GetDataDefinition

GetDataSetValues

Report

Each group of common data classes defined in IEC 61850-7-3 has a table like Table 12 to specify the services supported (or allowed).

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 13.

Table 13 – Trigger option

TrgOp	Semantics	Services allowed
dchg	data-change	A report or a log entry shall be generated due to a change of the value of the data attribute.
qchg	quality-change	A report or a log entry shall be generated due to a change of the value of the quality attribute.
dupd	data value update	A report or a log entry shall be generated due to freezing the value of a freezable attribute or updating the value of any other attribute. An updated value may have the same value as the old value.
Empty field	If the field is empty, then an application may use dchg or dupd to trigger reports or loggings	See for dchg or dupd respectively.

As depicted in Figure 64, the value of a data attribute that provides a specific **TrgOp** (trigger option) shall be monitored for reporting and logging if the report control block has enabled the specific trigger option (**TrgOps**). In the upper example of Figure 64, the **TrgOps** is **dchg**; the **TrgOp** of the data attribute is **dchg** for the first, **dupd** for the second, and **qchg** for the last data attribute. Reports are sent on data changes only, because only **dchg** is enabled in the report control block. In the second example, all changes will be reported. In addition, a report will be sent on the expiration of the integrity period.

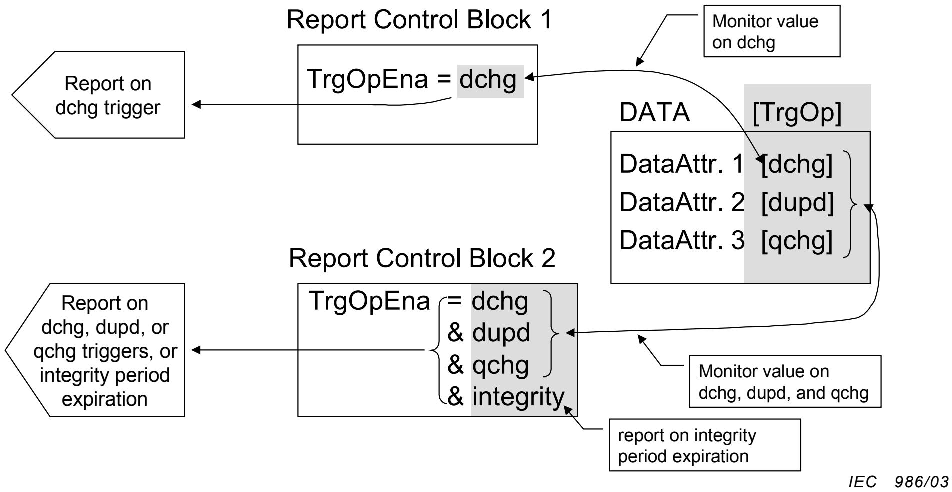


Figure 64 – Relation of TrgOp and Reporting

The column “value/value range” may contain enumerations (for example, stop | lower | higher | reserved); where “|” separates the values. The last column indicates if the attribute is mandatory, optional, conditional mandatory, or conditional optional.

13.2.3 Logical node class

Table 14 shows the table of a basic (foundation) logical node class defined in IEC 61850-7-2. The logical nodes contained in IEC 61850-7-4 inherit all definitions from this basic logical node class.

Table 14 – Logical node class (LN) definition

LOGICAL-NODE class		
Attribute name	**Attribute type**	**Explanation**
LNName	ObjectName	Instance name of an instance of LOGICAL-NODE
LNRef	ObjectReference	Path-name of an instance of LOGICAL-NODE
Data [1 to *n*]	DATA	
DataSet [0 to *n*]	DATA-SET	
BufferedReportControlBlock [0 to *n*]	BRCB	
UnbufferedReportControlBlock [0 to *n*]	URCB	
LogControlBlock [0 to *n*]	LCB	
If compatible LN class defined in IEC 61850-7-4 equals LLN0		
SettingGroupControlBlock [0 to 1]	SGCB	
Log [0 to 1]	LOG	
GOOSEControlBlock [0 to *n*]	GoCB	
GSSEControlBlock [0 to *n*]	GsCB	
MulticastSampledValueControlBlock [0 to *n*]	MSVCB	
UnicastSampledValueControlBlock [0 to *n*]	USVCB	
Services		
GetLogicalNodeDirectory GetAllDataValues		

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

The lines represent the attributes of the logical node.

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

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

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

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

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

The last six optional attributes are valid only for the logical node zero (**LLN0**). Exactly one logical node zero is defined to be contained in a logical device.

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

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

13.3 Service tables

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

Parameter name
Request
Parameter 1…
Parameter n
Response+
Parameter 1…
Parameter n
Response−

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

Usually the service table provides the request and response parameters of a specific service. Each parameter and the effect this parameter has on the processing of the service is described in this part of the standard in an abstract way. The sequences of the service primitives of confirmed services are depicted in Figure 65.

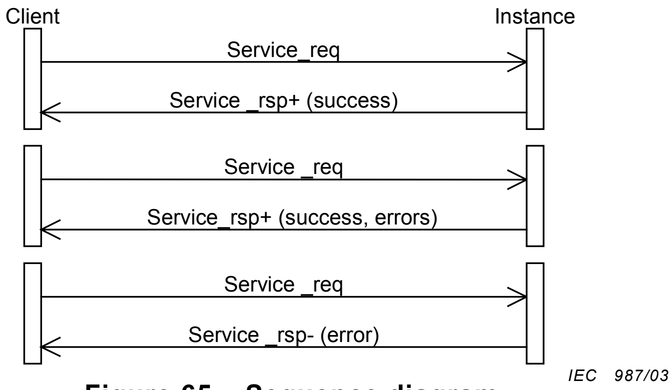


Figure 65 – Sequence diagram

13.4 Referencing instances

The standard differentiates between object names and object references. Object names (see Figure 66) identify an instance of a class at one hierarchy level (for example, "**Mod**" at the Data level or "**Q0XCBR1**" at the logical node level). "**Q0**" is a prefix and the "**1**" is a suffix to the name "**XCBR**". The concatenation of all the object names form the object reference (for example, "**MyLD/Q0XCBR1.Mode.stVal**").

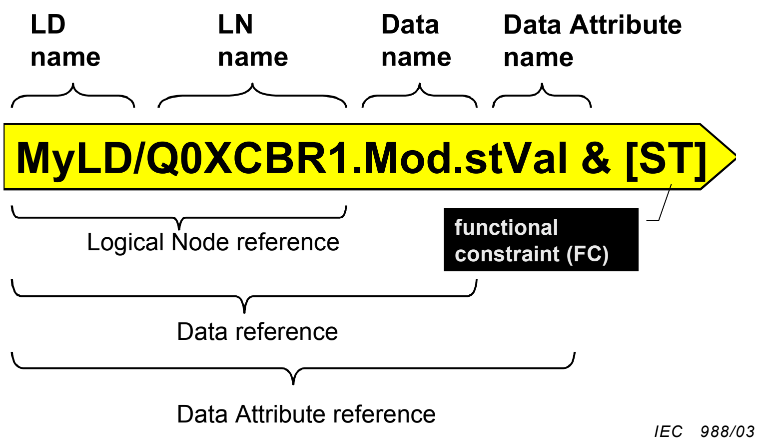


Figure 66 – References

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

The logical node name "**XCBR**" may be enhanced by a prefix (for example "**Q0**") and a suffix (for example "**1**") to build the logical node name ("**Q0XCBR1**"). The standardisation of prefixes and suffixes is outside the scope of this standard. All data names and data attribute names are used unchanged for instances; no prefixes and suffixes other than those defined in IEC 61850-7-4 shall be allowed for data names and data attribute names.

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

– Functionally constrained data (FCD), and
– Functionally constrained data attribute (FCDA).

The conceptual use of FCD and FCDA is shown in Figure 67 (using the MMS notation with "$" and the functional constraint (FC) between LNName and DataName).

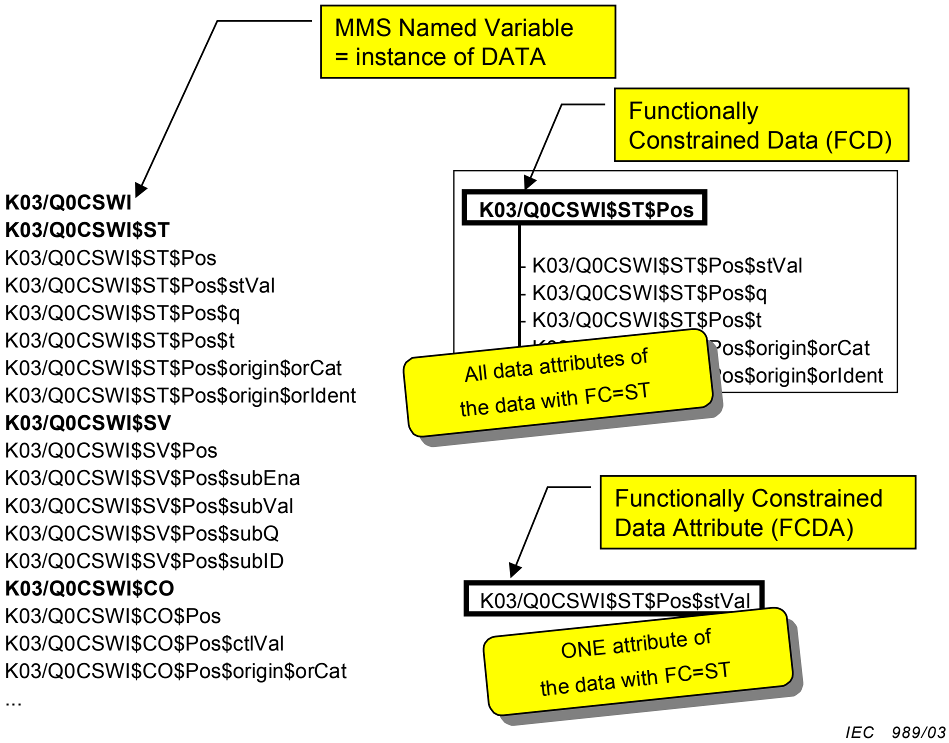


Figure 67 – Use of FCD and FCDA

Logical device names are not defined in the standard at all.

The logical device names introduced in a project will be described by the Substation Configuration Language (SCL).

The following example, an XML excerpt (according to IEC 61850-6) contains a substation section with one bay E1Q1, which contains a circuit breaker **QA1** and a disconnector **QB1**, both electrically connected together at node **L1**.

```
<Substation Ref="AB">
        <VoltageLevel Ref="E1">
                <Bay Ref="Q1">
                        <Device Ref="QA1" Type="CBR">
                                <Connection TNodeRef="L1"/>
                                <LNode Ref="XCBR1" LNClass="XCBR"/>
                        </Device>
                        <Device Ref="QB1" Type="DIS">
                                <Connection TNodeRef="L1"/>
                                <LNode Ref="XSWI2" LNClass="XSWI"/>
                        </Device>
                </Bay>
        </VoltageLevel>
</Substation>
```

The logical node reference may result in: "**AB.E1.Q1.QA1/XCBR1**" with "**AB.E1.Q1.QA1**" as the logical device name.

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

Figure 68 shows examples of object names and object references. The example at the top (first five lines) can be just five class definitions (not yet instantiated) or five instances of the classes "**E1.QA5/XCBR.Pos.ctlVal**", "...**stVal**", "...**q**", "...**t**", "...**ctlMode**". The object references do not in this case indicate if object references refer to classes or instances. The context where these references are used has to provide sufficient information to know what is meant (class or instance).

The other examples refer to instances only.

The LD name "**E1.QA5**" and its structure are outside the scope of the IEC 61850 series. The functional constraint (**FC**) is not shown in the object reference. The **FC** information may be mapped into the object reference in a SCSM; IEC 61850-8-1 maps the **FC** between logical node name and the data name ("**XCBR.CO.Pos**").

LD	LN	Data	DAttr.	FC	
E1.QA5	/XCBR	.Pos	.ctlVal	CO	class or instance
E1.QA5	/XCBR	.Pos	.stVal	ST	
E1.QA5	/XCBR	.Pos	.q	ST	
E1.QA5	/XCBR	.Pos	.t	ST	
E1.QA5	/XCBR	.Pos	.ctlModel	CF	
LD5	/YPTR2	.Temp	.mVal.i	MX	instance # 2
			.mVal.f	MX	
E1.QA5	/XCBR8	.Pos	.ctlVal	CO	instance # 8
E1.QA5	/XCBR8	.Pos	.stVal	ST	
E1.QA5	/XCBR8	.Pos	.q	ST	
E1.QA5	/XCBR8	.Pos	.t	ST	
E1.QA5	/XCBR8	.Pos	.ctlModel	CF	
Object Name	Object Name	Object Name	Object Name		

ObjectReference

IEC 990/03

Figure 68 – Object names and object reference

14 Name spaces

14.1 General

Each class defined in IEC 61850-7-4 (for example circuit breaker **LOGICAL-NODE**, **XCBR**), IEC 61850-7-3 (for example Double Point Status, **DPS**), and IEC 61850-7-2 (for example Buffered Report Control Block, **BRCB**) has a class name and a specific meaning associated with the class. Almost all classes are made up of other classes. The hierarchical class model of IEC 61850-7-x provides a naming hierarchy. Each name in the hierarchy has a semantic in the context where the name is used.

Figure 69 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 standardised 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.

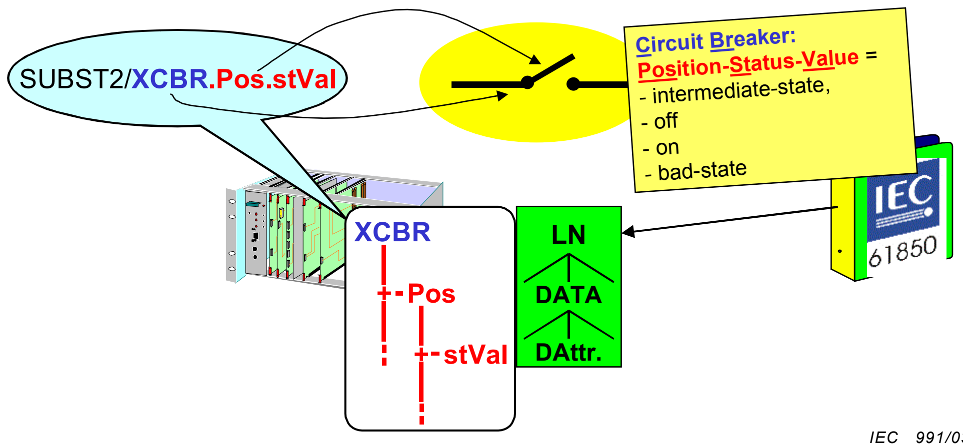


Figure 69 – Definition of names and semantics

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

For applications that need additional **LOGICAL-NODE**s, **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. Especially in the case where the same name, for example, **Pos** has been defined carrying different meanings, the standard needs to prevent a conflict with a single name having multiple definitions. Two meanings of a name of a **DATA** item are shown in Figure 70. The instance name **Pos** of a **DATA** item is used in the circuit breaker and in the nacelle of a wind turbine (**LOGICAL-NODE** with the name **WNAC**). In the context of a wind turbine, the position is defined as the plain angle of the nacelle. The value is a measured analogue in degrees (the SI Unit).

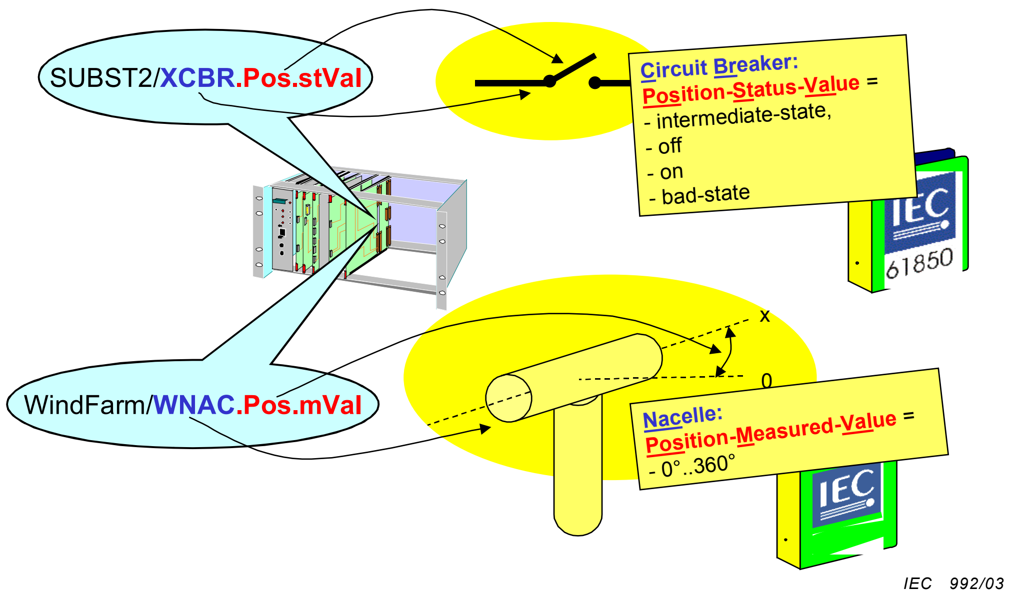


Figure 70 – One name with two meanings

The name **Pos** is used in two different contexts: substation (IEC 61850 series) and wind turbine (private or standardised specification).

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

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

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

14.2 Name spaces defined in IEC 61850-7-x

IEC 61850-7-4 and IEC 61850-7-3 define name spaces for application specific classes. IEC 61850-7-2 defines a names space for communication related (service) classes such as **BUFFERED-REPORT-CONTROL-BLOCK**, **LOG-CONTROL-BLOCK**, **LOGICAL-NODE**, **DATA**, **DATA-SET**.

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

As depicted in Figure 71, a name space is conceptually speaking a class repository containing various classes. A logical device is build up by instances derived from the classes of the repository. The standard class repository that comes with the IEC 61850 series is shown on the right hand side of Figure 71. An example of an additional name space is shown on the left hand side.

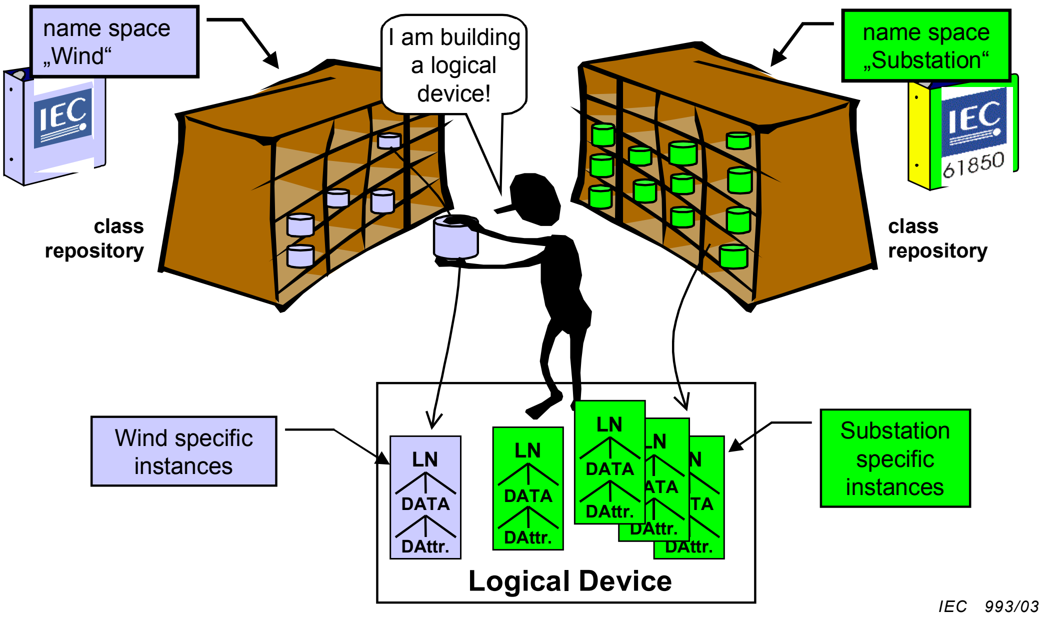


Figure 71 – Name space as class repository

The instances that are part of the logical device are coloured differently. The instances derived from the IEC 61850 series are green and the instance derived from the other standard is blue. As shown in Figure 72, the designation of the name space is represented by the attribute “logical device name space”:

ldNS = IEC 61850-7-4:2003

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

As long as all three documents (IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2) are of the same edition, it is sufficient to refer only to the IEC 61850-7-4. IEC 61850-7-4 has normative references which apply to the other two documents. The underlying instances of **LOGICAL-NODE**s, **DATA**, and **DATA-ATTRIBUTE**s have an implicit name space (i.e., IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2) which is derived by the normative references in IEC 61850-7-4.

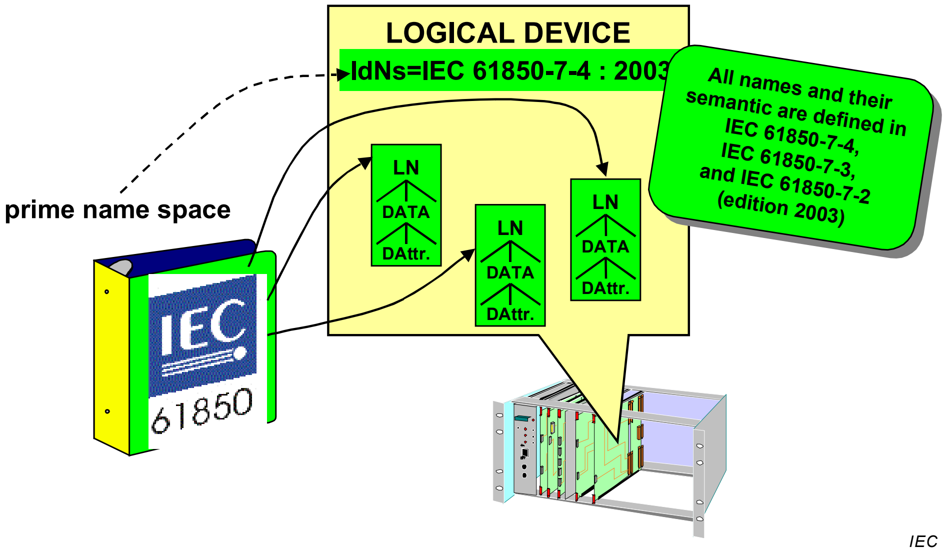


Figure 72 – All instances derived from classes in a single name space

As soon as additional class definitions are used in an instance of that class, the logical device shall provide information about the additional semantic. The example instances of Figure 73 are derived from three different name spaces: IEC 61850-7-4, the other standard document, and a private name space (Vendor ABC). Since the majority of instances are derived from IEC 61850-7-4, the logical device name space **ldNs** is still IEC 61850-7-4 (prime name space).

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

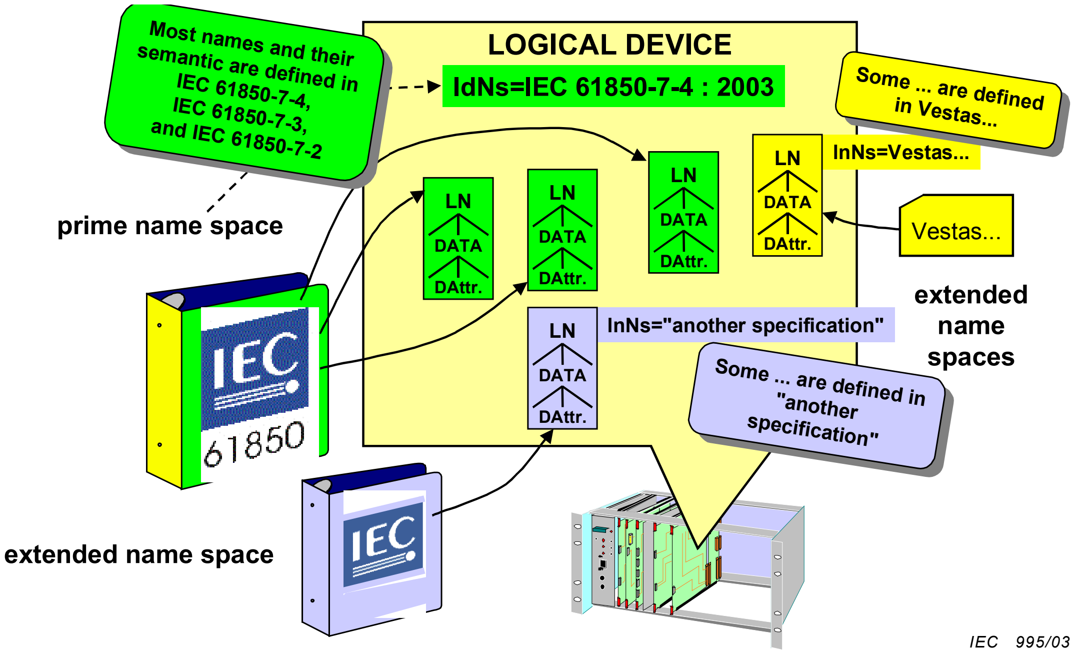


Figure 73 – Instances derived from multiple name spaces

The logical devices name space could be "IEC 61850-7-4:2003" or any other name space depending on the context in which a logical device is defined and used. The example given in Figure 74 shows how another specification could for example inherit all classes (LOGICAL-NODE, DATA, and common DATA classes) from IEC 61850-7-4 and IEC 61850-7-3. In that case the other specification would define a superset name space. Since the basic sets from different standards or other definitions are maintained completely independently from each other, superset name spaces shall be avoided to minimise the risk of inconsistencies and not to endanger interoperability.

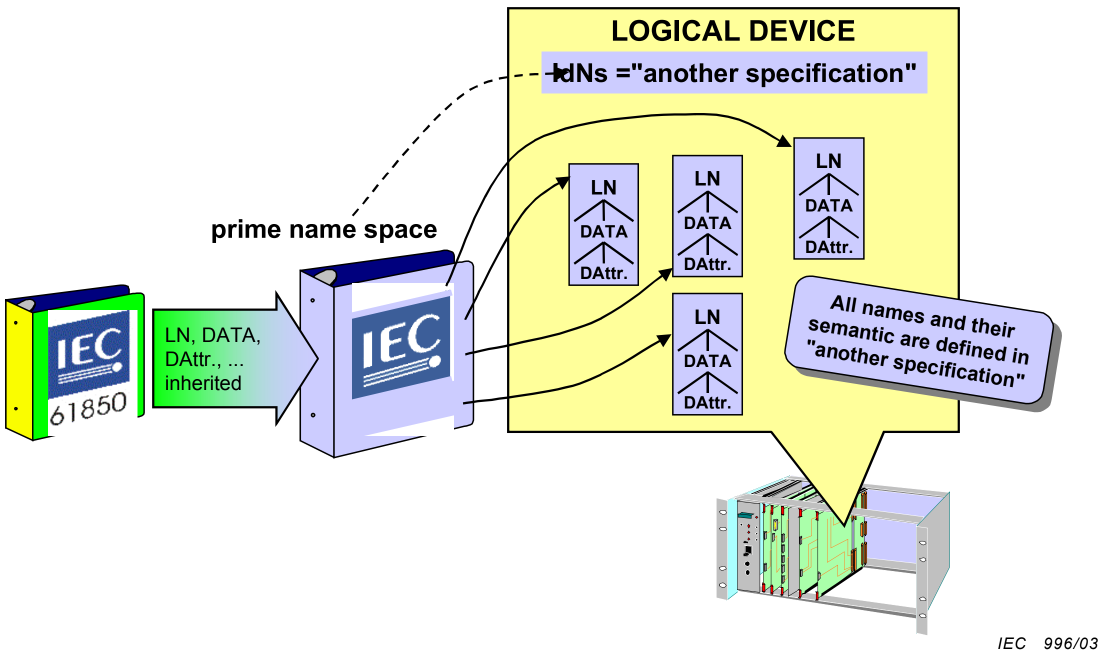


Figure 74 – Inherited name spaces

The **ldNs** has the value "other specification: year of publication". Since all classes are contained in that single name space, the underlying instances have the implicitly the same name space. They do not need to have an explicit value for their name spaces.

14.3 Specification of name spaces

14.3.1 General

The following three name spaces shall be defined to provide sufficient information to allow the unambiguous interpretation of the instances of the classes **LOGICAL-NODE**, **DATA**, **DATA-SET**, and the various control blocks, for example, BRCB:

a) **Logical node name space** shall be a technical specification containing the definition of the **LOGICAL-NODEs** (including the underlying classes and services) defined for a specific application domain (for example, substation and feeder devices),

b) **Data name space** shall be a technical specification containing the definition of the **DATA** class (including the underlying definitions and services) defined for a specific application domain (for example, substation and feeder devices), and

c) **Common data class name space** shall be a technical specification containing the definition of the common **DATA** classes (including the underlying definitions and services) defined for a specific application domain (for example, substation and feeder devices).

If the class definition is a specialisation of another class – the basic class – then the visible string for name space names shall be structured by the following concatenation:

name of new name space > name of basic name space > ...> ...

The “>” shall be a reserved symbol indicating the specialisation.

It is recommended to use the same table notation as in IEC 61850-7-4, IEC 61850-7-3, and IEC 61850-7-2. The name spaces may be defined in a single or in multiple documents.

14.3.2 Definition of logical node name space

The logical node name space shall be the specification that contains (and possibly references) all semantic definitions of all **LOGICAL-NODE** classes (and its underlying classes) defined for a specific application domain.

In the case of the IEC 61850 series, the logical node name space contains the following specifications:

IEC 61850-7-4 (**LOGICAL-NODE**, for example, MMXU, and **DATA**, for example, **PhV**, **A**, **W**, **PF**),

IEC 61850-7-3 (common **DATA** classes, for example, **WYE** for **PhV**) by reference, and

IEC 61850-7-2 (all classes) by reference,

including the year of the edition.

An example of logical node and data name spaces is shown in Figure 75.

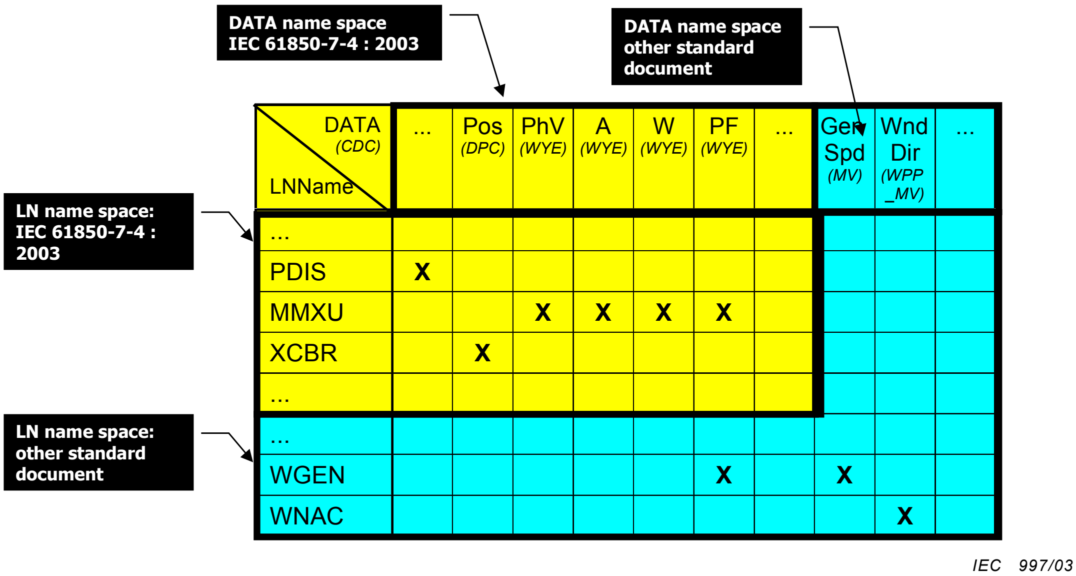


Figure 75 – Example of logical node and data name spaces

The logical node name space shall contain the name of the **LOGICAL-NODE** and the **DATA** which are part of the **LOGICAL-NODE**.

14.3.3 Definition of data name space

The data name space shall be the specification that contains (and possibly references) all semantic definitions of all **DATA** classes (and its underlying DataAttributes) defined for a specific application domain.

In the case of the IEC 61850 series, the data name space contains the following specifications:

IEC 61850-7-4 (**DATA** for example, **PhV**, **A**, **W**, **PF**),

IEC 61850-7-3 (common **DATA** classes, for example, **WYE** for **PhV**) by reference, and

IEC 61850-7-2 (all classes) by reference,

including the year of the edition.

The data name space contains the name of the **DATA** and the common data class which shall be used to create the DataAttributes of the instance of **DATA**.

14.3.4 Definition of common data class name space

The common data class name space shall be the specification that contains all semantic definitions of all common **DATA** classes defined for a specific application domain.

In the case of the IEC 61850 series the common data class name space contains the following specifications:

IEC 61850-7-3 (common **DATA** classes, for example, **WYE** with DataAttributes, for example, **ctlVal**, **q**, **PhsA**) by reference, and

IEC 61850-7-2 (all classes) by reference,

including the year of the edition.

An example of common data class name spaces is shown in Figure 76.

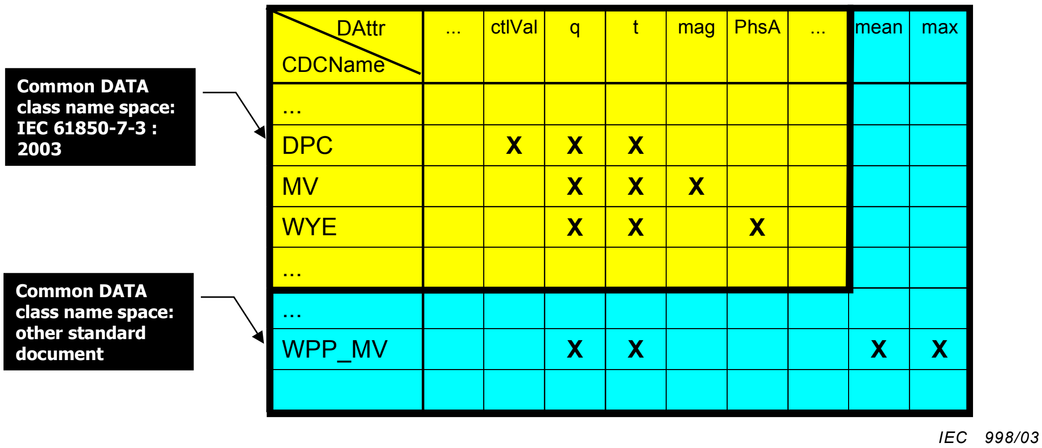


Figure 76 – Example common data class name spaces

The common data class name space shall contain the names of the common data classes and the DataAttributes, for example, **ctlVal**, **q**, and **PhsA** which shall be used to create the DataAttributes of the instance of **DATA**.

14.4 Attributes for references to name spaces

14.4.1 General

The following four attributes that reference name spaces are defined:

1) **Attribute logical device name space** (**ldNs**) shall reference the prime technical specification used for the whole logical device.
2) **Attribute logical node name space** (**lnNs**) shall reference the logical node name space of a single instance of **LOGICAL-NODE**.
3) **Attribute data name space** (**dataNs**) shall reference the data name space of a single instance of **DATA**.
4) **Attribute common data class name space** (**cdcNs**) shall reference the CDC name space of the CDC used for the definition of a single instance of **DATA**.

The common data classes contain the DataAttributes **ldNs** and **lnNs** as shown in the excerpt of Table 15.

NOTE 1 The conditions in the last column of Table 15 are defined in IEC 61850-7-3.

Table 15 – Excerpt of logical node name plate common data class (LPL)

LPL class					
Attribute name	**Attribute type**	**FC**	**TrgOp**	**Value/value range**	**M/O/C**
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
				configuration, description and extension	
...	...	...	...	...	...
ldNs	VISIBLE STRING255	EX		shall be included in **LLN0** only; for example “IEC 61850-7-4:2003”	AC_LN0_M
lnNs	VISIBLE STRING255	EX			AC_DLD_M

The common data classes contain the DataAttributes **cdcNs** and **dataNs** as shown in the excerpt of Table 16.

Table 16 – Excerpt of common data class

Applied by all common data classes					
Attribute name	**Attribute type**	**FC**	**TrgOp**	**Value/value range**	**M/O/C**
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
...	...	...	...	...	...
configuration, description and extension					
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M

NOTE 2 The conditions in the last column of Table 16 are defined in IEC 61850-7-3.

14.4.2 Attribute for logical device name space (ldNs)

The IEC 61850 series does not define standard logical devices; i.e., no logical device name is defined. The DataAttribute logical device name space **ldNs** shall be used to indicate which technical specification has been used as the prime name space for the whole logical device. If the underlying logical node name spaces are identical with the name space contained in the **ldNs**, then the logical node name space need not to be contained in the individual logical nodes.

NOTE The **ldNs** can be understood as a substitute for the **lnNs** in all or many underlying logical nodes.

The attribute **ldNs** shall be a DataAttribute of the name plate **NamPlt** of the **LOGICAL-NODE-ZERO** (**LLN0**). The attribute **ldNs** shall be as defined in the common **DATA** class **LPL** (logical node name plate) of IEC 61850-7-3.

For the application of the first edition of IEC 61850-7-x as the prime name space, the value shall be “IEC 61850-7-4:2003”.

The attribute **ldNs** shall be available in every **LOGICAL-NODE-ZERO** (**LLN0**).

The ObjectReference for the DataAttribute **ldNs** shall be:

LDName/LLN0.NamPlt.ldNs

14.4.3 Attribute for logical node name space (lnNs)

The DataAttribute logical node name space **lnNs** shall be used to indicate which technical specification has been used as the name space for a specific logical node. The attribute shall be available only if the logical node name space deviates from the name space referenced in the attribute **ldNs** of the **LLN0**.

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

For the application of the first edition of IEC 61850-7-4 as the logical node name space, the value shall be “IEC 61850-7-4:2003”.

The ObjectReference for the DataAttribute **lnNs** shall be:

LDName/LNName.NamPlt.lnNs

14.4.4 Attribute for data name space (dataNs)

The DataAttribute data name space **dataNs** shall be used to indicate which technical specification has been used as the name space for a specific data. The attribute shall be available only if the data name space deviates from the name space referenced in the attribute **lnNs** of the logical node to which the data belongs.

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.

For the application of the first edition of IEC 61850-7-4 as the data name space, the value shall be “IEC 61850-7-4:2003”.

The ObjectReference for the DataAttribute **dataNs** shall be:

LDName/LNName.DataName[.DataName[. ...]].dataNs

14.4.5 Attribute for common data class name space (cdcNs)

The DataAttribute common data class name space **cdcNs** shall be used to indicate which technical specification has been used as the common data class name space for the creation of a specific data. The attribute shall be available only if the common data class name space deviates from the name space referenced in the attribute **lnNs** of the logical node to which the data belongs.

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

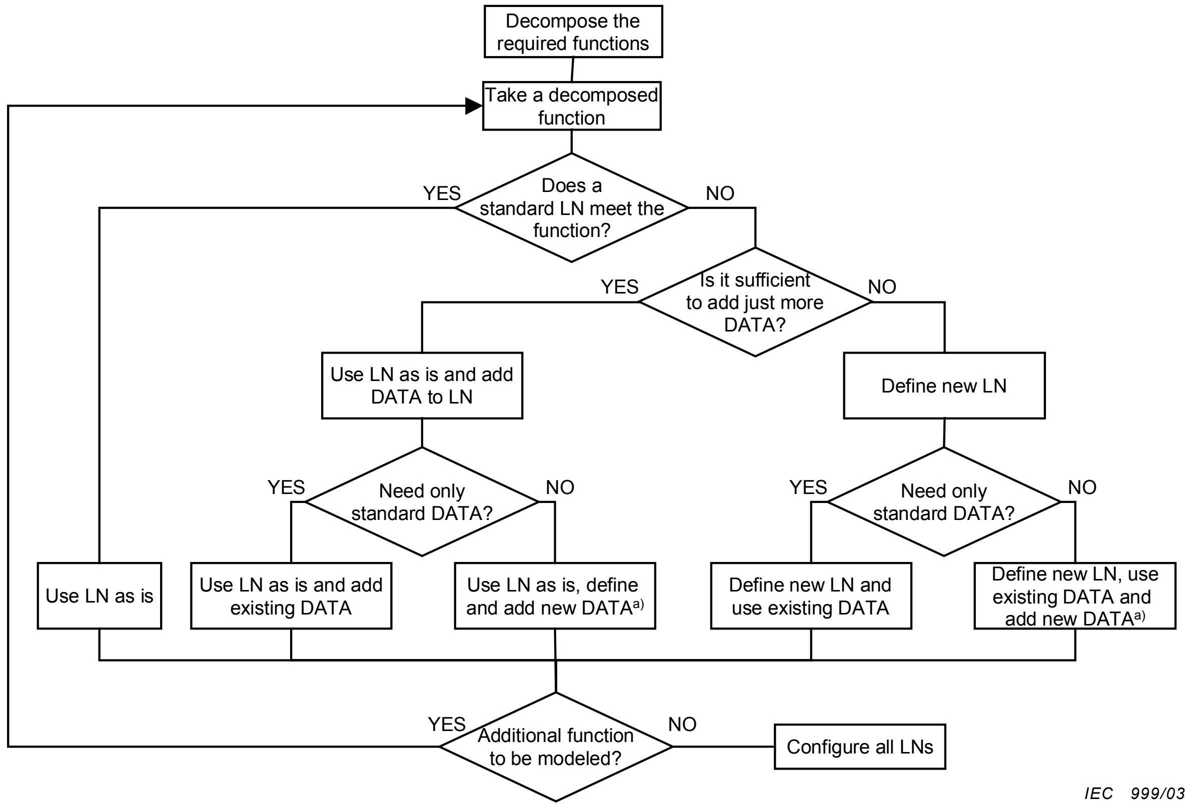
For the application of the first edition of IEC 61850-7-3 as the common data class name space, the value shall be “IEC 61850-7-3:2003”.

The ObjectReference for the DataAttribute **cdcNs** shall be:

LDName/LNName.DataName[.DataName[. ...]].cdcNs

14.5 Common rules for extensions of name spaces

In Annex A of IEC 61850-7-4, strong and comprehensive rules for extensions are given. These extension rules are reflected in the extension rules for name spaces. The name spaces can be extended with the rule depicted in the conceptual diagram shown in Figure 77.



a) New DATA based on existing or new CDC.

Figure 77 – Extensions of name spaces (conceptual)

Building **LOGICAL-DEVICEs** is a stepwise process with the following steps:

a) Decompose the required application function to a degree of granularity of the existing logical node classes, or start from the existing set of logical node classes of your application domain, i.e. substation in case of the IEC 61850 series that are listed in IEC 61850-7-4,

b) chose a **LOGICAL-NODE** from the existing (may be standardised) **LOGICAL-NODEs** if the **LOGICAL-NODE** meet the requirement,

c) chose a **LOGICAL-NODE** from the existing (may be standardised) **LOGICAL-NODEs** and add new **DATA** if the **LOGICAL-NODE** meets the core requirements,

d) define a new **LOGICAL-NODE** if the **LOGICAL-NODE** under b) and c) do not meet the requirements,

   – the new **LOGICAL-NODE** may contain **DATA** that are already defined in other **LOGICAL-NODEs**, or

   – the new **DATA** may be defined using existing or new common **DATA** classes (CDC),

e) if required, assign a application specific LN-prefix and LN-instance-ID to the instance of LOGICAL-NODE,

f) repeat steps b) to f) for the other functions,

g) configure the **LOGICAL-NODE**s.

The use of a standardised name space (IEC 61850-7-4) and an extended data and a common data class name space (ABB_2273) is shown in Figure 78.

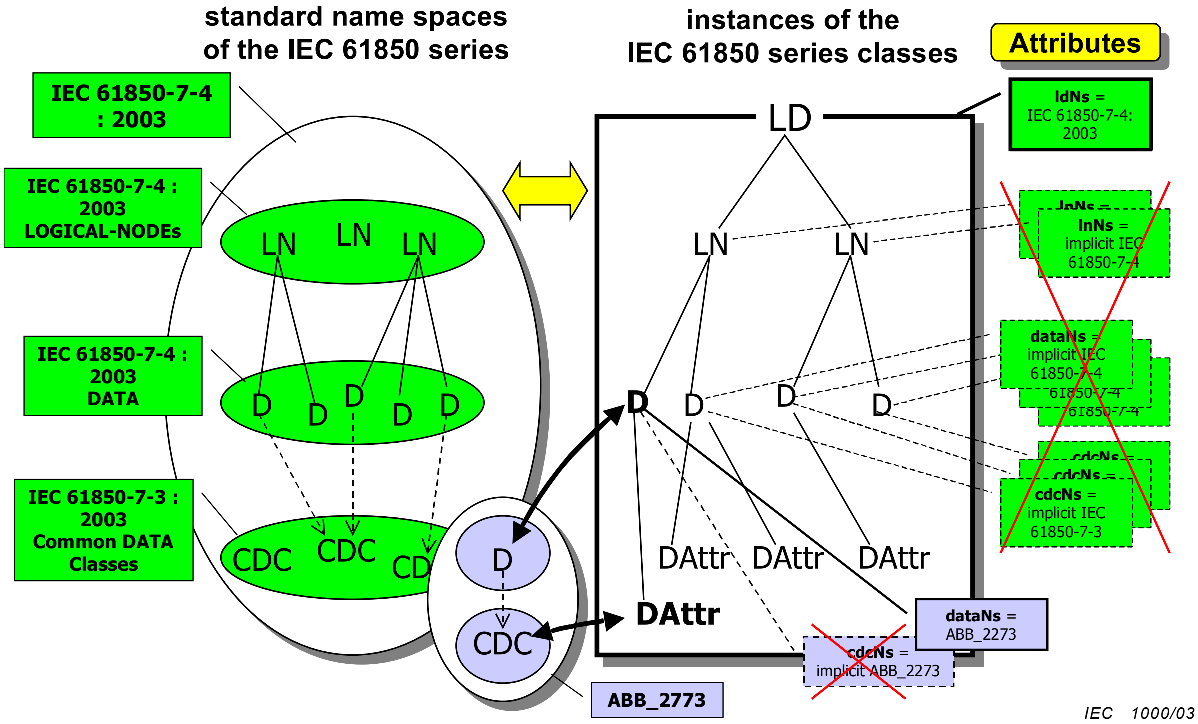


Figure 78 – Use of extended name space (conceptual)

The instance of **DATA** on the left hand side in the logical device is derived from the technical specification referenced by the name "ABB_2773". This new name space contains just one data and one common data class. This extension (relative to the other name spaces which are defined in IEC 61850-7-4) is marked in the instance of data with the attribute **dataNs** = "ABB_2773". The common data class name space used for that data does not need to be marked in the instance of that data because it is implicitly given in the name space definition of "ABB_2773".

For the unambiguous interpretation of the semantic of the logical device, only two marks are required: the **ldNs** (IEC 61850-7-4:2003) and the **dataNs** (ABB_2773) for the extended definition.

15 Approaches for the definition of a new semantic

15.1 General

The definition of a new semantic could follow various approaches. The decision concerning which approach best meets the requirements depends on the application area and mainly on the question of which parts of the definition should be reused in other definitions.

NOTE The examples used are simplified. They show the information required to demonstrate the different approaches. In a real specification, more details are required. The examples have been arbitrarily selected.

To demonstrate the different approaches, three examples are discussed. The first approach defines a fixed semantic, the second defines a more flexible approach that allows the configuration of two attributes, and the third is based on the definition of a new common data class. The new common data class simplifies the definition of the logical node tremendously because the table of the logical node just lists the process data. The configuration-specific information is hidden in the logical node. The new common data class can be reused for any other definition of data which need the same DataAttributes.

The following example of a requirement has been chosen to demonstrate the variety of approaches.

15.2 Semantic for new definition

The objective of the new definition (used to explain the different approaches) is to define statistical DATA for the mean outside temperature of a substation. The **DATA** shall provide mean values for 1 hour and for 10 minutes. For simplicity of the example, the logical node that contains the **DATA** is not shown but it will contribute to the semantic of the **DATA**. Examples may be the existing **MSTAT** related to metering statistics or a new LN such as **MENV** (Measuring Environmental Values).

15.3 Approach 1 (fixed semantic)

The names of the measurand **DATA** are defined as: **TmpMean60** and **TmpMean10**.

The type of the measurand **DATA** (the common data type) is defined as: **MV** (Measured Value).

Logical node: WXYZ	
Data	**common data class**
TmpMean60	MV
TmpMean10	MV

Semantic for **TmpMean60**: the mean value at the full hour (00:00, 01:00, etc.).

Semantic for **TmpMean10**: the mean value at the full hour and n × 10 minutes after the hour (00:00, 00:10, 00:20, etc.).

15.4 Approach 2 (flexible semantic)

The names of the measurand **DATA** are defined as: **TmpMean1** and **TmpMean2**.

The type of the measurand **DATA** (the common data type) is defined as: **MV**.

There are four accompanying configuration **DATA** defined that are used to set the time interval (indicated by “**I**”) and start time (indicated by “**S**”) for each of the measurands.

The names of the configuration **DATA** are defined as: **ITmpMean1** and **STmpMean1**, and **ITmpMean2** and **STmpMean2**.

The type of the configuration **DATA** (the common data type) is defined as: **ASG** (Analogue Setting).

Logical node: WXYZ	
Data	**Common data class**
TmpMean1	MV
TmpMean2	MV
ITmpMean1	ASG
STmpMean1	ASG
ITmpMean2	ASG
STmpMean2	ASG

Semantic for **TmpMean1**: the mean value as per setpoint values.

Semantic for **TmpMean2**: the mean value as per setpoint values.

15.5 Approach 3 (reusable flexible semantic)

The names of the measurand **DATA** are defined as: **TmpMean1** and **TmpMean2**.

The type of the measurand **DATA** (the common data type) is defined as: **MVStat**.

Logical node: WXYZ	
Data	**Common data class**
TmpMean1	MVStat
TmpMean2	MVStat

A new common data class has been defined which can be reused for many statistical measured values.

Common data class: MVStat	
Data attribute	**Attribute type**
statVal	AV
Int	INT32
strtTm	TimeStamp

Annex A
(informative)

Overview of IEC 61850-7-x, IEC 61850-8-x, and IEC 61850-9-x

A.1 Introduction

The modelling and implementation methods applied in the different parts of the standard and their relation are shown in Figure A.1. As depicted on the left-hand side of Figure A.1, this part of IEC 61850 defines the basic principles and modelling methods.

The compatible data and logical node object classes, and the common data classes and attributes are defined in IEC 61850-7-4 and IEC 61850-7-3.

NOTE Since these classes are defined for substation and feeder equipment, there may be other object classes defined for various other application domains within or outside the scope of IEC Technical Committee 57. They are relevant for Figure A.1 only if they are built according to the approach of the IEC 61850 series.

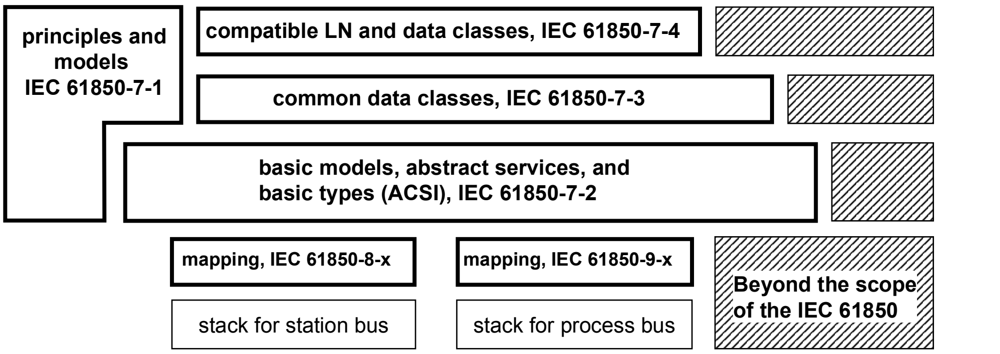


Figure A.1 – Overall communication system architecture

In order to be able to handle these different interrelated aspects, the whole system is decomposed in smaller components. The top-down approach resulted in the following documents:

- IEC 61850-7-4 Compatible logical node classes and data classes (several hundred terms and unique names).
- IEC 61850-7-3 Common data classes (common details about the content of terms defined in IEC 61850-7-4).
- IEC 61850-7-2 Abstract communication service interface (ACSI) (common service class models with services and parameters to communicate with instances of the classes of IEC 61850-7-4 and IEC 61850-7-3).
- IEC 61850-8-1 Specific communication service mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-1 and ISO/IEC 9506-2) and to ISO/IEC 8802-3 (encoding of data, services, and service parameter).
- IEC 61850-9-1 Specific communication system mappings (SCSM) – Sampled values over serial unidirectional multidrop point to point link (encoding of data, services, and service parameters).
- IEC 61850-9-2 Specific communication system mappings (SCSM) – Sampled values over ISO/IEC 8802-3 (encoding of data, services, and service parameters).
- IEC 61850-6 Substation automation system configuration language (representation of all optional data of IEC 61850-7-4 and IEC 61850-7-3).

A.2 Compatible logical node classes and data classes (IEC 61850-7-4)

A.2.1 List of LN groups (IEC 61850-7-4)

A list of all groups of logical nodes is contained in Table 2 of this part of IEC 61850.

A.2.2 LN classes (IEC 61850-7-4)

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

A.2.3 Data classes (IEC 61850-7-4)

A total number of some 500 data classes are defined in IEC 61850-7-4. Table A.1 shows an excerpt of a few **DATA** classes with their names and semantic definitions.

The data names are composed using standardised abbreviations listed in a table at the beginning of IEC 61850-7-4 (some 260 abbreviations are defined), 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.

Table A.1 – Excerpt of data classes for measurands

Data name	Semantics
AcsCtlFail	Number of access control failures detected.
AcuPaDsch	Acoustic level of partial discharge in db.
AgeRat	Ageing rate, for example of transformer.
Alm	General single alarm.
AlmLstOv	TRUE = indication that the alarm list has overflowed.
AlmThm	Thermal alarm.
AlmVal	Alarm value is the pre-set value for a measurand that when reached will result in an alarm.
Amp	Current of a non-three-phase circuit.
Ang	Angle between phase voltage and current.
AngCor	Phase angle correction of a phasor (used for example for instrument transformers).
AngInd	This data indicates the check result of the differences between the angles of the busbar and line voltages. FALSE indicates that the angle difference is below the required limit. The angle difference criteria for the synchronising are fulfilled. TRUE indicates the angle difference exceeds the limit. The synchronising process shall be aborted because the angle criteria are not fulfilled (synchrocheck) or shall be continued with turbine control activities (synchronising).
...	...

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

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

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

Table A.2 – List of common data classes

Common data class specifications for status information
Single Point Status (SPS)
Double Point Status (DPS)
Integer status (INS)
Status Indication Group (SIG)
Integer Status (ISI)
Protection Activation Information (ACT)
Directional protection activation information (ACD)
Security Violation Counting (SEC)
Binary Counter Reading (BCR)
Common Data Class Specifications for Measurand Information
Measured Value (MV)
Complex measured value (CMV)
Sampled value (SAV)
WYE
Delta (DEL)
Sequence (SEQ)
Harmonic Value for WYE (HVWYE)
Harmonic Value for DEL (HDEL)
Common Data Class Specifications for Controllable Status Information
Controllable Single Point (SPC)
Controllable Double Point (DPC)
Controllable Integer Status (INC)
Binary Controlled Step Position Information (BSC)
Integer Controlled Step Position Information (ISC)
Common Data Class Specifications for Controllable Analogue Information
Controllable analogue set point information (APC)
Common data class specifications for status settings
Single point setting (SPG)
Integer status setting (ING)
Common data class specifications for analogue settings
Analogue setting (ASG)
Setting curve (CURVE)
Common data class specifications for description information
Device name plate (DPL)
Logical node name plate (LPL)
Curve shape description (CSD)

Annex B
(informative)

Allocation of data to logical nodes

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

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

– Data consisting of instantaneous values of current and voltage are assigned to the logical nodes “current transformer” and “voltage transformer” respectively.

– Data consisting of calculated values of current and voltage, for example root-mean-square r.m.s., are assigned to the logical node “measurement unit”.

– Data consisting of voltage and step position are assigned to the logical node “tap changer controller”. Same holds for the tap change commands.

– Data consisting of a (fault) impedance *Z* are assigned to the logical node “distance protection”. The same applies to the protection trip.

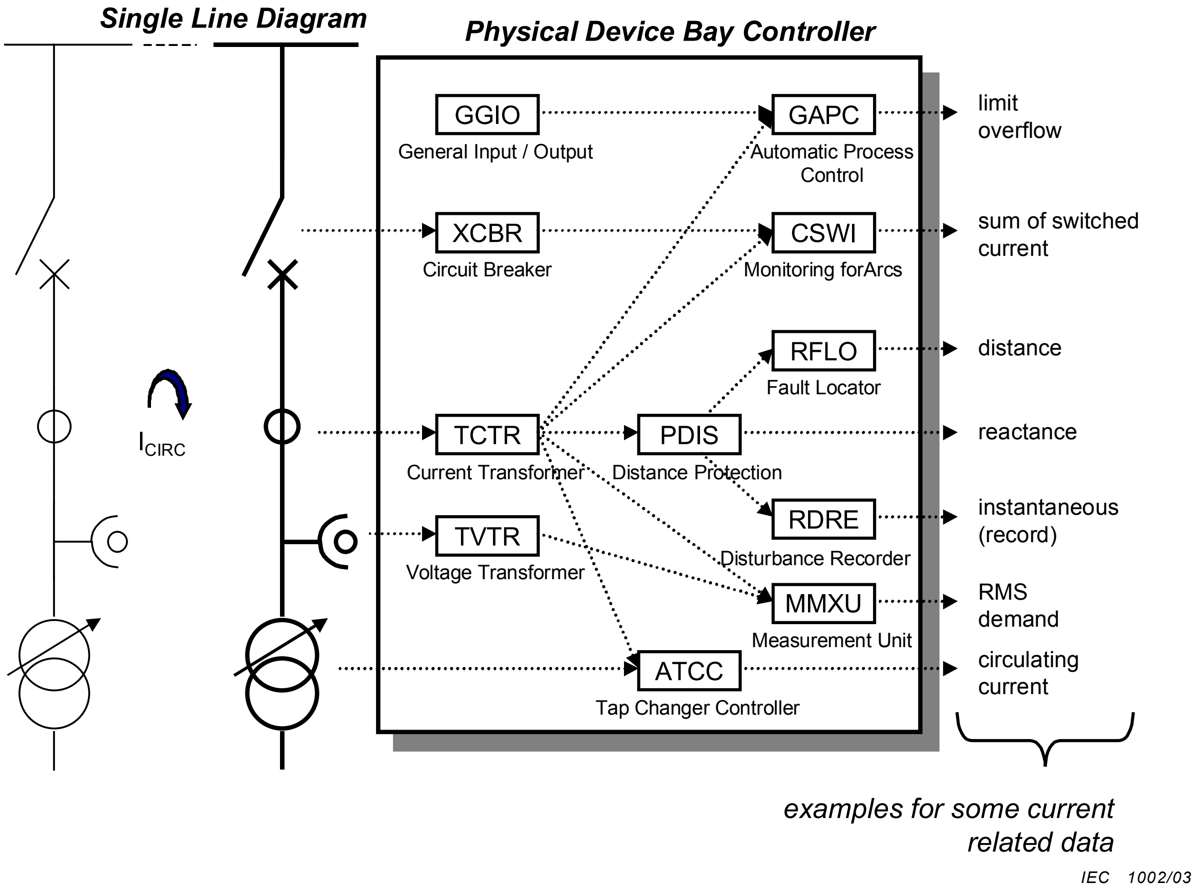


Figure B.1 – Example for control and protection LNs combined in one physical device

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

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

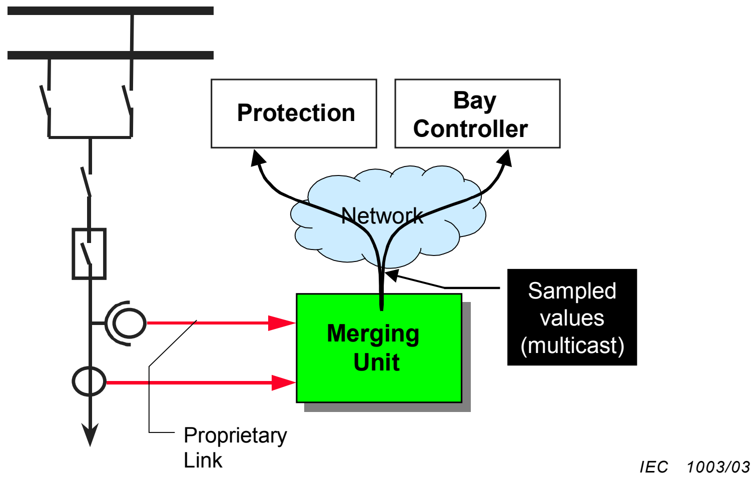


Figure B.2 – Merging unit and sampled value exchange (topology)

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

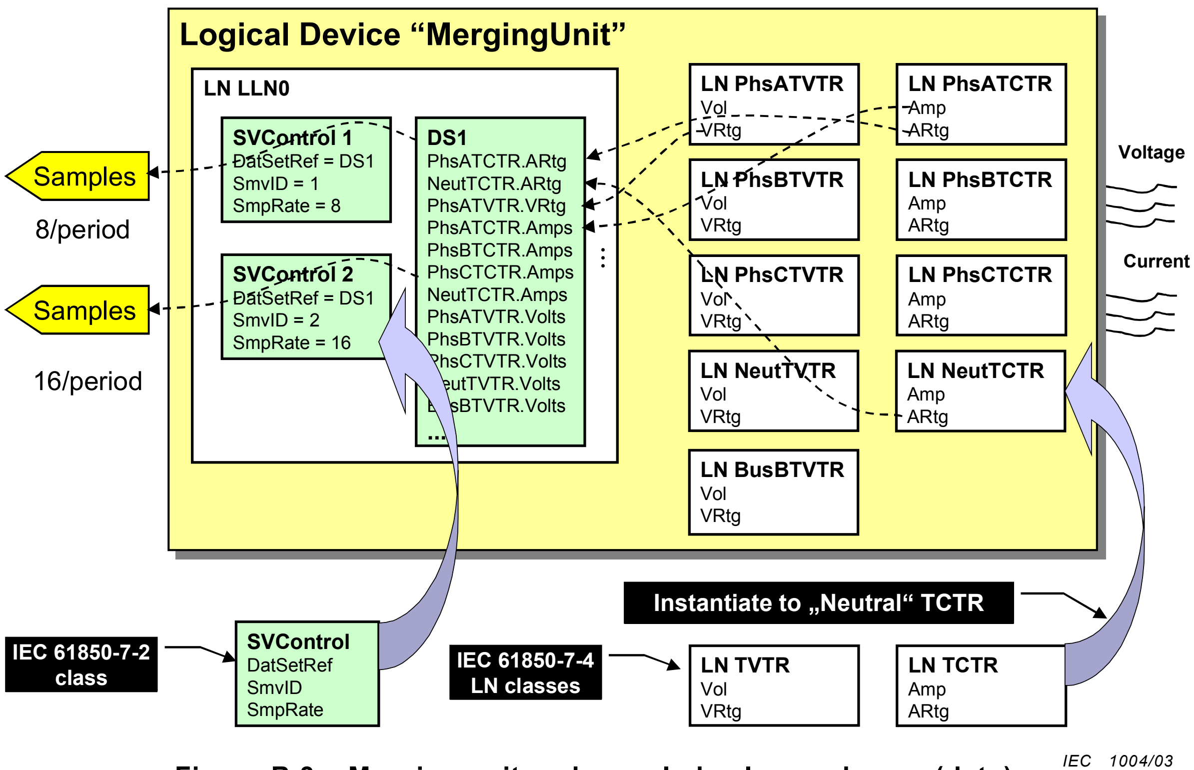


Figure B.3 – Merging unit and sampled value exchange (data)

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

Current transformer – **TCTR** class in IEC 61850-7-4 instantiated for three phases and neutral:

PhsATCTR, PhsBTCTR, PhsCTCTR, NeutTCTR,

Voltage transformer – **TVTR** class in IEC 61850-7-4 instantiated for three phases, neutral, and bus bar:

PhsATVTR, PhsBTVTR, PhsCTVTR, NeutTVTR, BusBTVTR,

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

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

Annex C
(informative)

Use of the substation configuration language (SCL)

C.1 Introduction

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

C.2 SCL and options in logical nodes

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

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

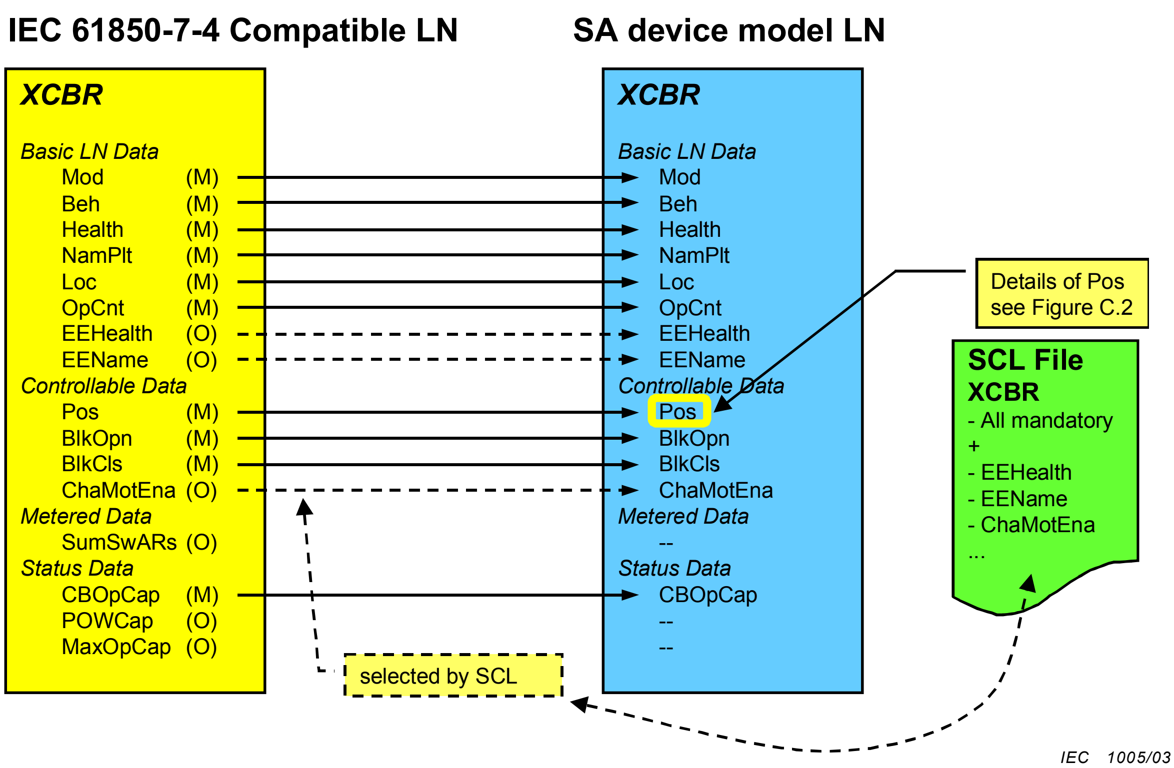


Figure C.1 – Application of SCL for LNs (conceptual)

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

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

C.3 SCL and options in data

At the logical node level, it is sufficient to list the names of the data which are optional. The SCL for the data requires the list of optional data attributes as well as the initialisation (configuration) values for several data attributes.

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

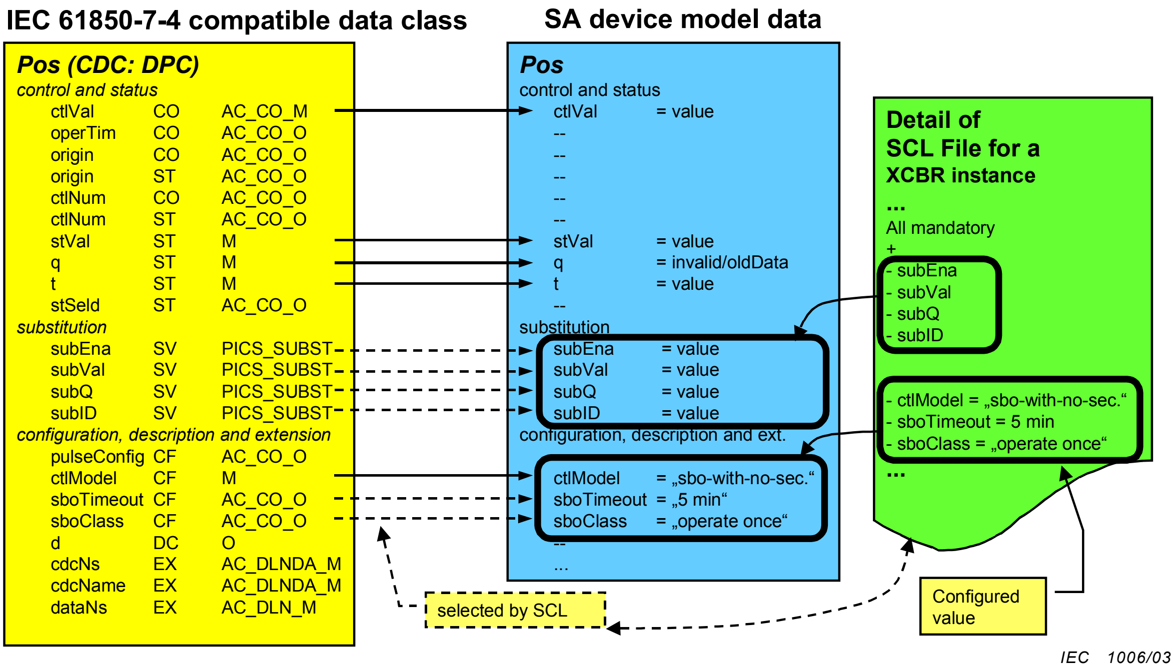


Figure C.2 – Application of SCL for data (conceptual)

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

Annex D
(informative)

Applying the LN concept to options for future extensions

D.1 Introduction – Seamless telecontrol communication architecture

The seamless communication as depicted in Figure D.1 allows the modelling of the control center view (CC view) of a substation and the communication between substations and control centers.

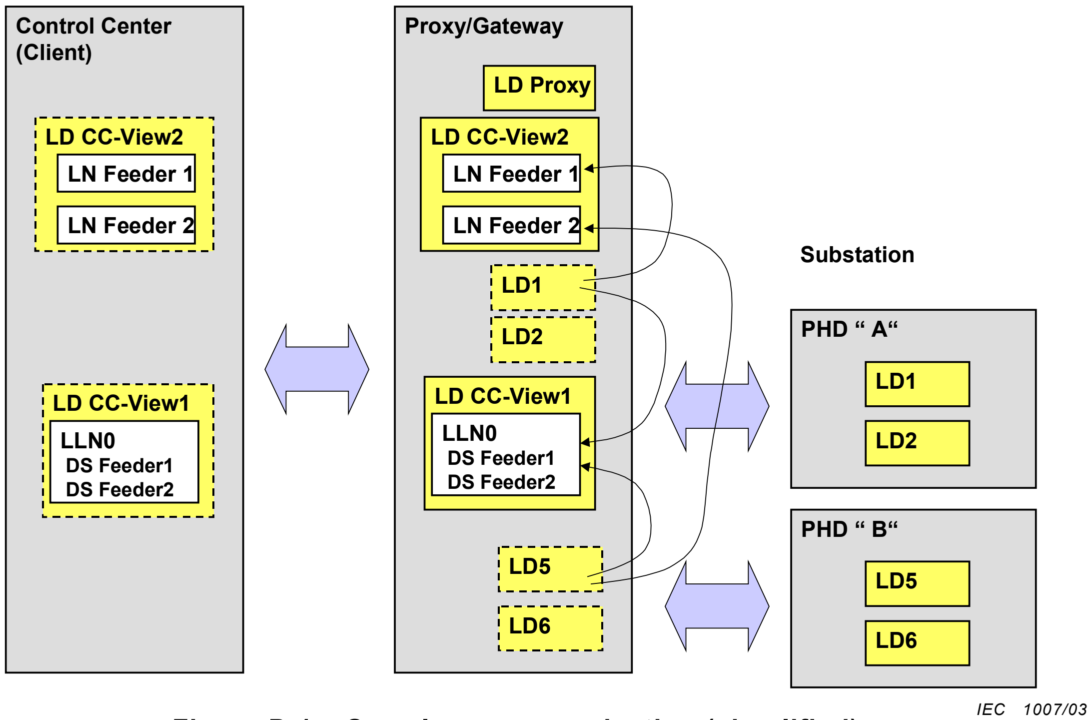


Figure D.1 – Seamless communication (simplified)

The control center can access the substation through a physical device serving as a gateway. This provides several options to access the data of the substation:

1. The gateway/proxy provides the capability for a direct access to the physical devices in the substation.
2. Within the gateway/proxy, a logical device (for example CC-View1) may define data sets that collect the information required in the control center.
3. As an alternate solution, new logical node classes and data classes may be defined to be used in the gateway/proxy that provide the control center view of the substation (in the example above, instantiated in the logical device CC-View2).

If new logical node classes and data classes are to be defined to reflect the control center view, this shall be harmonized with the CIM model in particular with regard to the names of the data classes.

This provides three options to access the data of the substation:

Option 1 is most valuable for maintenance issues.

Option 2 could be used for operational issues, but requires expensive engineering effort.

Option 3 seems to be a promising solution for operational issues, since the same engineering concept as used within the substation could be applied for the whole control system of a utility. Therefore, this option was further investigated.

Combinations of options are possible. For example it may be suitable to use option 3 for operational purposes and option 1 for engineering and maintenance purposes.

Example for the requirement of a new logical nodes:

For the control center, the building blocks of the substation are bays (for example feeders or transformer bays). Therefore, a new logical node group “bay” might be defined with logical nodes for different bay types. These logical nodes with their data classes will provide the control center view of the substation.

An example is given in Figure D.2.

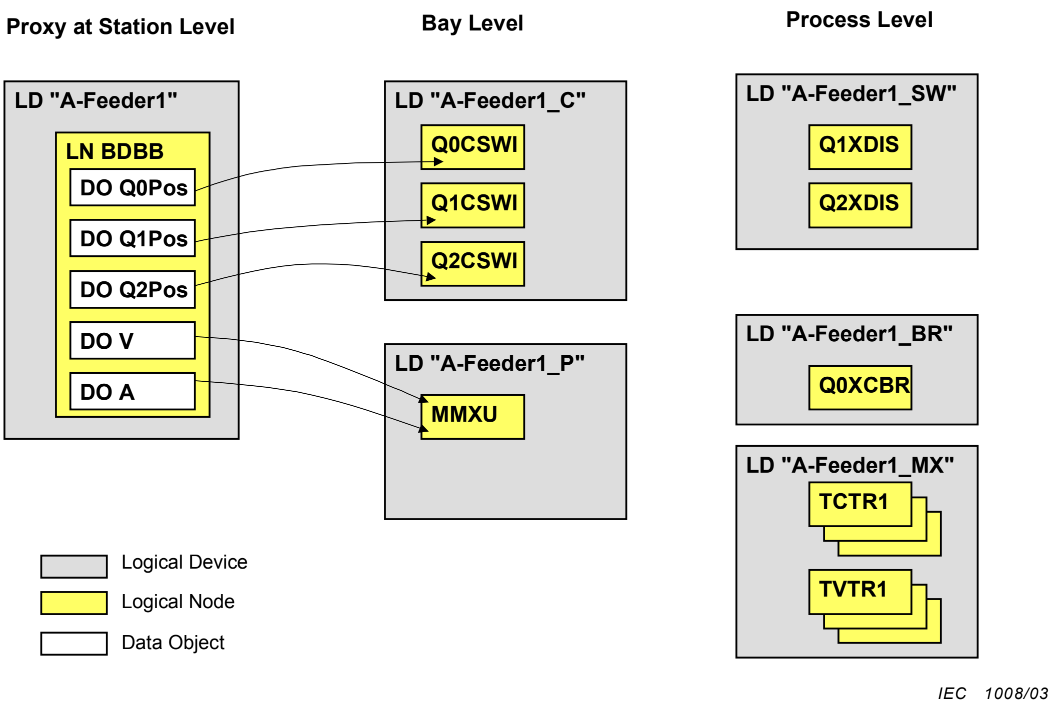


Figure D.2 – Example for new logical nodes

This supports a similar modelling approach for the control center as for the substation. As a consequence, the same engineering concepts and tools as well as the same communication software can be used.

EXAMPLE

Logical node of group "bay":

BDBB double busbar
BHCB one and a half CB

Logical devices defined, for example, per voltage level:

SSAtlanta_110
SSAtlanta_380

Some data objects:

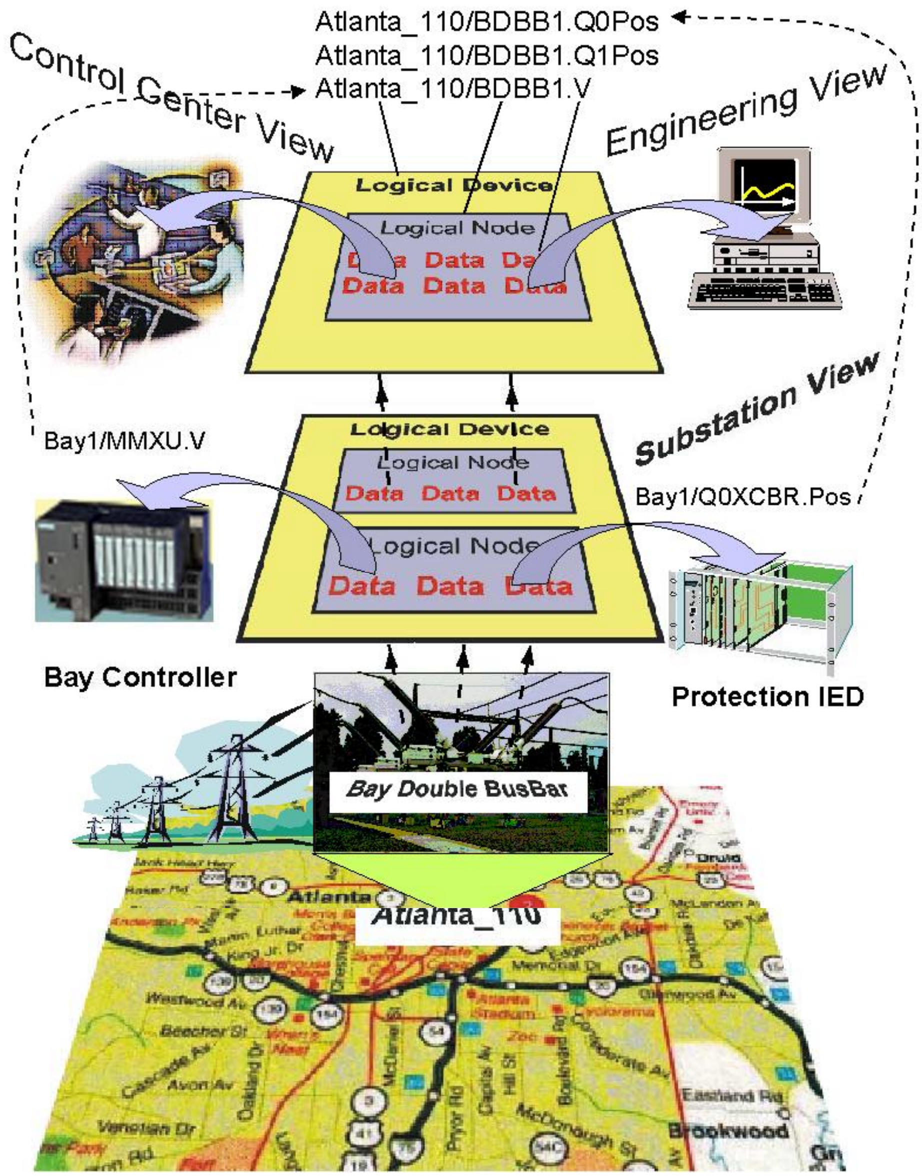
SSAtlanta_110/BDBB1.Q0Pos
SSAtlanta_110/BDBB1.Q1Pos
SSAtlanta_110/BDBB1.V
SSAtlanta_110/BDBB2.Q0Pos
SSAtlanta_110/BDBB2.Q1Pos
SSAtlanta_110/BDBB2.V
SSAtlanta_380/BHCB1.QAPos
SSAtlanta_380/BHCB1.QBPos
SSAtlanta_380/BHCB1.QCPos

Basically, the LN Bxxx is another virtual view on the same real world object. How the LN Bxxx receives the information is an implementation issue. It may for example subscribe to reports from LN Q0CSWI or it may directly forward a control command to Q0CSWI.

The data objects in the new LNs will be of the same common data classes (including important metadata) as the original data objects. However, in a first approach only the mandatory attributes may be supported for the seamless communication to the control center. By creating new logical device and logical node instances dedicated for a specific control center view, the names may be defined to meet the system operator's preferences. The name translation is done in the gateway/proxy of the substation.

The new LNs may also define new data classes such as summary alarms representing a logical combination of individual alarms.

An example is depicted in Figure D.3. The substation view can be mapped to one or more control center views. In the example above, the logical device SSAtlanta_110 could be the view of the control center A and the logical device SSAtlanta_380, the view of control center B.



IEC 1009/03

Figure D.3 – Example for control center view and mapping to substation view

D.2 Teleprotection

D.2.1 Distance protection

In case of the selective isolation of a faulted line the distance protection (logical node PDIS) exchanges signals for “release” or “block”. These signals should be the same as used inside the substation. The LN **PSCH** (line protection scheme) is already part of the standard. The data model should be the same, i.e. according to the IEC 61850 series, as well as the mapping and stack selection. For special communication requirements, a mapping with different layers 1 and 2 of the ISO/OSI reference model may be applied.

D.2.2 Differential protection

The line differential protection exchanges the same signals (mostly samples or phasors) as the differential protection (**PDIS**) inside the substation like transformer differential or busbar differential protection. The data model should be the same, i.e. according to IEC 61850, also the mapping and stack selection. For special communication requirements, a mapping with different layers 1 and 2 of the ISO/OSI reference model may be applied.

D.2.3 Extended functionality

Using the complete approach to the IEC 61850 series to teleprotection (line protection) would also in the future allow the use of very distributed functions, for example interlocking with the inclusion of the switch position at the other side of the line as well.

Annex E
(informative)

Relation between logical nodes and PICOMs

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

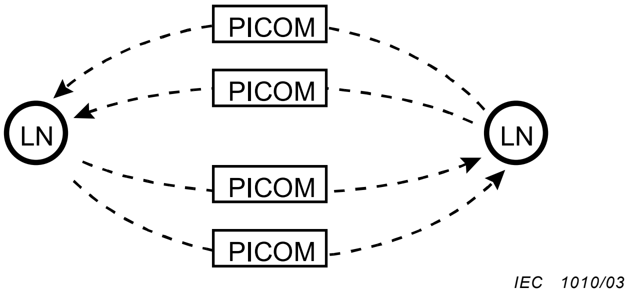


Figure E.1 – Exchanged data between subfunctions (logical nodes)

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

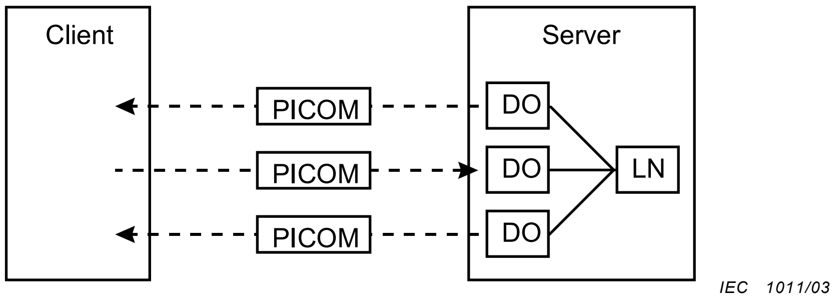


Figure E.2 – Relationship between PICOMS and client/server model

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

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

Annex F
(informative)

Relation between IEC 61850-7-x (IEC 61850-8-x) and UCA 2.0®[4]

Figure F.1 depicts the general relation between various parts of the IEC 61850 series and UCA®.

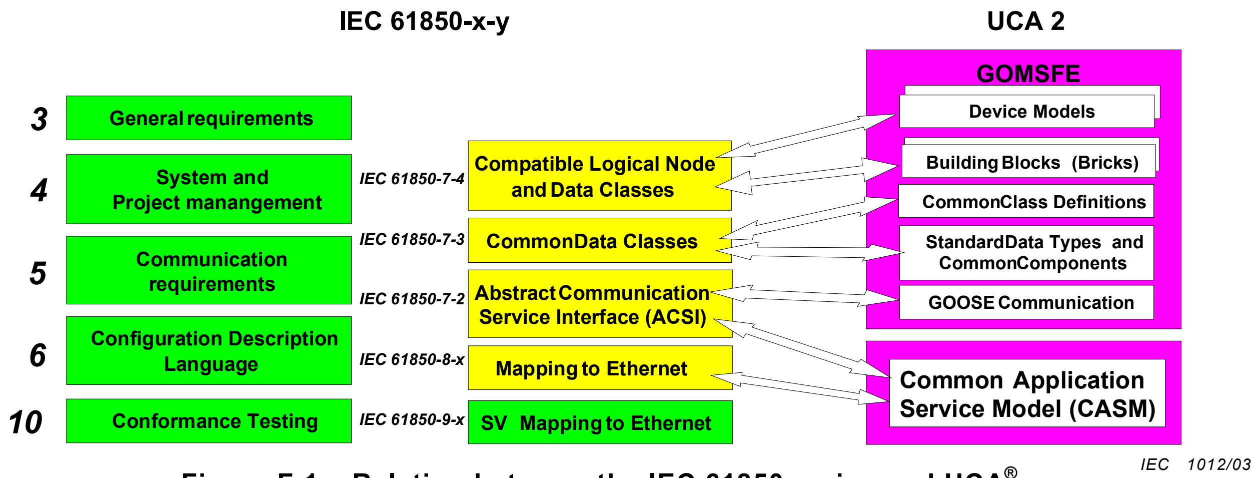


Figure F.1 – Relation between the IEC 61850 series and UCA®

4 UCA® is a registered trademark of EPRI, Palo Alto, (USA).

Bibliography

IEEE-SA TR 1550, 1999: Utility Communications Architecture (UCA®) Version 2 [5].

[5] UCA® is a registered trade mark of EPRI, Palo Alto (USA).

Index

Standards Survey

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

Customer Service Centre (CSC)

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

or

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

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

A Prioritaire

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

RÉPONSE PAYÉE

SUISSE

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

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

....

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

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

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

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

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

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

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

- not at all ❑
- nearly ❑
- fairly well ❑
- exactly ❑

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

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

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

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

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

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

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

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

ISBN 2-8318-7064-X

ICS 33.200

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND