

# A Layered Hierarchical Object-Oriented View of IEC 61850

Iony Patriota de Siqueira  
CHESF  
iony@tecnix.com.br  
Brazil

## Abstract

A top-down approach is presented for the object-oriented modeling of the standard IEC 61850. Paralleling the traditional seven layers of the OSI (Open Systems Interconnection) model for network communication, this paper develops a unified model of all major software classes defined by IEC 61850, grouped in stacked layers, ranging from the higher slices seen from users and integrators, to the lower layers interfacing network communication and middleware. These models compose four concurrent views, as seen from different stakeholders, that summarize the architecture of standard IEC 61850.

## Key-words

Object-Oriented Model, UML, SCL, XML, IEC 61850, Substation Automation Systems.

## Introduction

One of the difficulties related to IEC 61850 is its huge size and complexity. Comprising more than 1000 printed pages of textual and graphic contents, the standard has defied the comprehension of protection and automation engineers, from manufactures to utilities and integrators. Since its first edition, focusing mainly on substations, new functionalities and logical nodes have been added, extending its application to domains like wind power, hydropower, distributed energy, inter substation communication, etc. New additions are planned for condition monitoring, testing, logics, etc., and harmonization with other standards.

Lacking of a digital version has also created difficulties for tool makers, integrators and software developers, generating differences and doubts in its interpretation, and reducing the interoperability of models compiled by different tools. This scenario has motivated the Technical Committee 57 of IEC to plan a digitalized version of IEC 61850, to be available on-line as a machine readable model of the standard. This model will ease the work of relay and software developers, but will not reduce the difficulty faced by automation and protection engineers to understand and apply the standard. Teachers and consultants will continue to have hard work to explain its main aspects to students and practicing engineers, due to its size, novelty and complexity.

A traditional approach to understand a complex process is to conceptualize it in hierarchical views, allowing a progressive modeling with increasing levels of details. To apply this method to IEC 61850 and to ease its understanding and widespread use, this paper presents a layered UML (Unified Modeling Language) model of all abstract classes defined by the standard. The model mimics the organization of the OSI communication model, presenting a sliced top-down view of its standardized objects, organized in layers of class packages. Each layer is supported by lower level layers, while supporting upper level layers, mimicking a hierarchical organized middleware service ranging from low level message classes, to high level SCL (Substation Configuration Language) classes. This modeling principle allows an overall view of all abstract classes, draught in a single

UML class diagram suitable for a teaching poster in a classroom, or as a personal guide to the whole standard.

The models are formed by classes organized in seven layers (by coincidence the same number of the OSI levels), also called models, comprising, from top to bottom, the Substation Model, the Product Model, the Type Model, the Communication Model, the Functional Model, the Data Model and the Service Model. The upper level, or Substation Model, is formed by all SCL classes used to describe a Substation Automation System, with its external and inter-substation communication. These classes are expanded and interfaced to lower level classes, increasing the model detail in terms of data type, communication and services. The deepest layer comprises all low level service and message classes, like Client-Server, Goose and Sampled-Value messages, used to interface with a network middleware. These layers are described using UML views and class diagrams.

### UML Views

Unified Modeling Language (UML), as defined by the OMG (Object Managing Group), is a graphical language for specifying, visualizing, constructing and documenting software systems. As a modeling language, it allows the user to define several models of a system, as distinct views of the same subject or business area, according to the user interests. In UML, a model captures a view of a system, with a specific purpose. It is a complete description of those aspects relevant to the purpose of the model, at the appropriate level of detail. Models can be used to give different views of a system to different stakeholders, to focus on singular aspects of a system at a time, or to express the results of different stages in a software development process. To define a model one needs to define its purpose, give a complete picture of the system, while omitting irrelevant information. Usually, four generic views are usual to model a business or technical project in UML, named as the Strategic, Structural, Functional and Process Views, in this case, of a Substation Automation System (SAS), as shown in Figure 1.

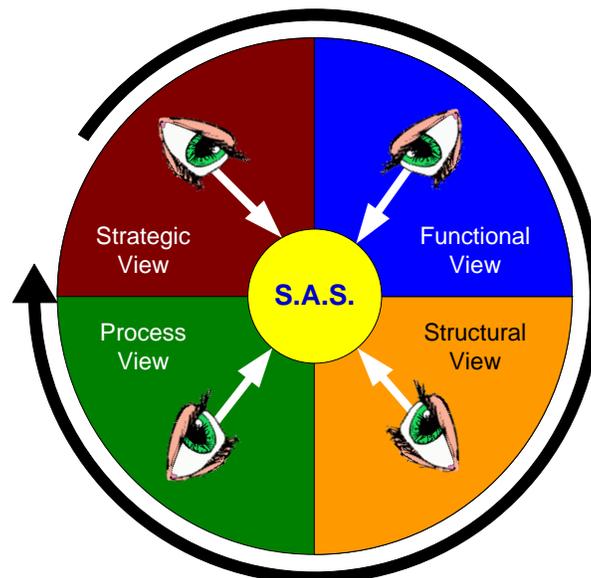


Figure 1 – Different Views of a Substation Automation System

The views are usually built in a sequential process, shown in the clockwise direction on Figure 1, starting from a Strategic View. When applied to an SAS project, a Strategic (or Business) View contains an external/integrator or owner/stakeholder perspective and conceptual model of the

overall organization and objectives of the design. Clients and software toolmakers also benefit from this external view to model user requirements for a new project. This is further detailed in a Functional View, with the logic and operational blocks that form the main SAS functions to attain the strategic objectives. Users, operators and maintainers are among the stakeholders that use this operational view to manage the substation. A further detailed perspective is supplied next by the Structural View, with the building pieces of static objects and data that shape each functional block. Manufacturers and IED suppliers are responsible for the data structure supported by the devices used on the substation supporting interoperability. Finally, the Process View details the methods, services, dynamic data and messages exchanged among the building blocks and the external environment to attain the desired SAS functions. Messages classes are modeled as a sub package of the Service Package that form this view, or as part of the middleware used to support the communication among IEDs, being specific to each application. The environment, in an SAS context, may be all electrical, mechanical and process equipment in the substation or power plant, local and remote operators and control centers, etc. Field engineers, networkers and communication personnel are responsible to supply the network and middleware to support this view. If these views are implemented in IEC 61850, they use instantiated objects from a collection of seven packages of classes, defined on the standard and named and numbered in this paper according to the following models:

1. Substation Model;
2. Product Model;
3. Type Model;
4. Communication Model;
5. Functional Model;
6. Data Model; and
7. Service Model.

These packages can be associated and organized in a hierarchical layered structure, according to the four UML views defined above, with their corresponding stakeholders, as shown on Figure 2, using the above package numbering.

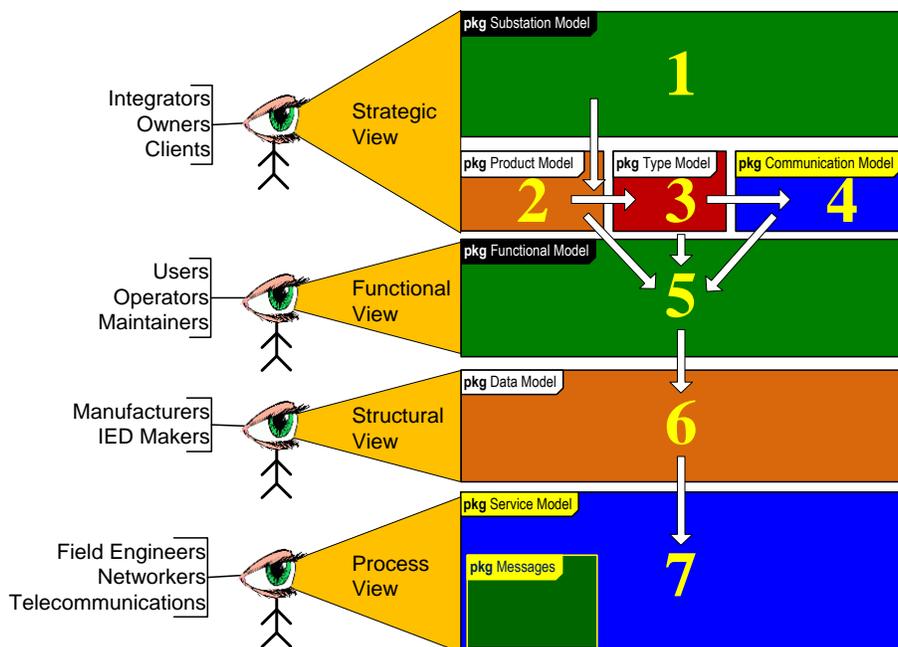


Figure 2 – Hierarchical Layered Organization of SAS Views and Models

A Strategic View of the SAS can be built with instantiated classes from the Substation, Product, Type and Communication Models, and specified in SCL, the XML (eXtensible Markup Language) based Substation Configuration Language of IEC 61850. This is followed by a Functional View of the SAS, built from objects instantiated from the Functional Model defined by IEC 61850, mainly using logical nodes. The next Structural View details all data objects used by logical nodes and functional objects, inherited from classes defined in the IEC 61850 Data Model. Finally, the Process View collects all objects derived from classes defined on the Service Model and its Messages package to support the methods of the above layers. These views, as collections of packages of classes, are organized by hierarchical layered dependency relations defined in a unique UML class diagram shown in Figure 16 on the Annex of this paper, and described in the subsequent paragraphs. The packages in this picture follow the same numbering of Figure 2.

### Substation Model

The Substation Model is an object hierarchy based on the functional structure of the substation that uses the Substation Configuration Language to describe a Strategic View of an SAS. It links the automation functions to substation functions and equipment, and derives functional designations for the logical nodes from the substation structure. Its objects are instantiated from the SCL abstract classes Substation, VoltageLevel, Bay, Equipment, SubEquipment, ConnectivityNode and Terminal. These classes are related in the UML class diagram in Figure 3, composing the Substation Model package, part of the Strategic View at the upper Layer 1 of the general UML model of Figures 2 and 16. IEC 61850 documentation contains the detailed description of these classes, in this and the following packages.

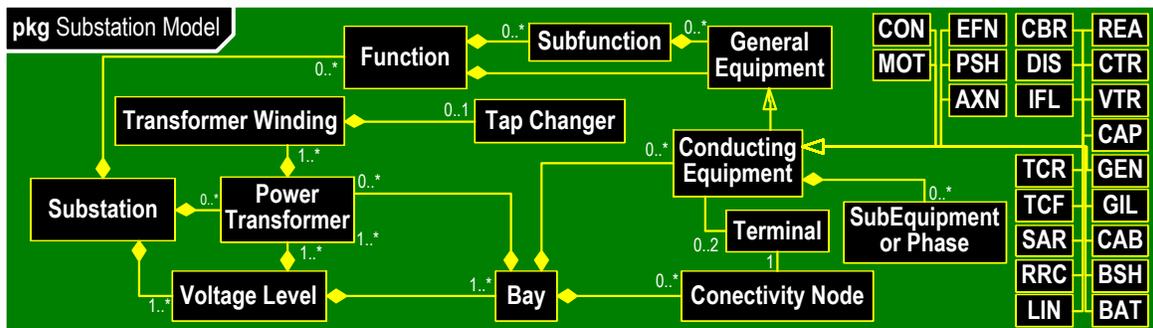


Figure 3 – Substation Model Package

Using these classes a substation can be modeled physically, by VoltageLevels containing electrically connected substation Bays, as part of the switchyard with instantiated Equipments and physical apparatus; or functionally, by instantiated Equipments performing Subfunctions of the main Functions attained by the Substation. These are linked by ConnectivityNode and ConductingEquipments objects between primary devices and Terminals, and by electrical connection points of a primary apparatus at a single voltage level. Figure 4 shows a hierarchical composition of a typical substation, with a physical and functional model of the items instantiated from these classes.

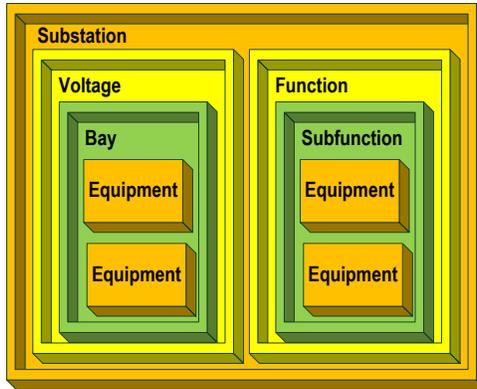


Figure 4 – Typical Substation Instantiated Objects

These objects are implemented using classes defined in a Product Model, in the second layer of the integrated model, as part of the Strategic or Business View of an SAS.

### Product Model

The Product Model is used to specify the characteristics of all IEDs (Intelligent Electronic Devices) as part of a Strategic View of the SAS. It describes the configuration of each IED, with its access points, logical devices and logical nodes. It also defines the capabilities of each IED by the communication services offered and default or configuration values. The following abstract classes are described in IEC 61850 for the Product Model: IED, Server, LDevice, LNode, LN0, Router and Clock. These classes are related in the UML class diagram of Figure 5, forming the Product Model package, part of the Strategic View on Layer 2 of the general UML layered model of Figure 16.

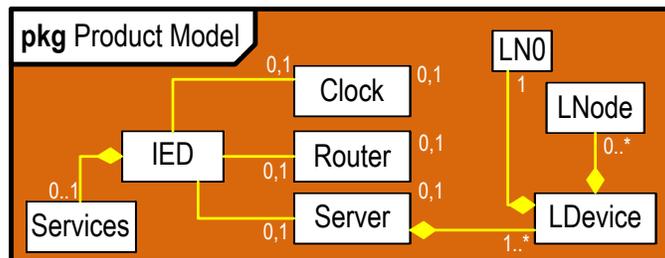


Figure 5 – Product Model Package

IEDs are substation automation (SA) devices performing SA functions by means of logical nodes (LNs) that reside in Servers, the communication entity within an IED. Within a Server, an LDevice models a logical device (LD) composed of LNodes, or logical nodes (LN) that contains objects carrying their data. Routers, a component of a communication network, connects different sub networks, blocks real-time messages outside the standard scope, while Clocks indicate where a sub network master clock is located. Figure 6 shows a general composition of a typical Product Model with these instantiated objects.

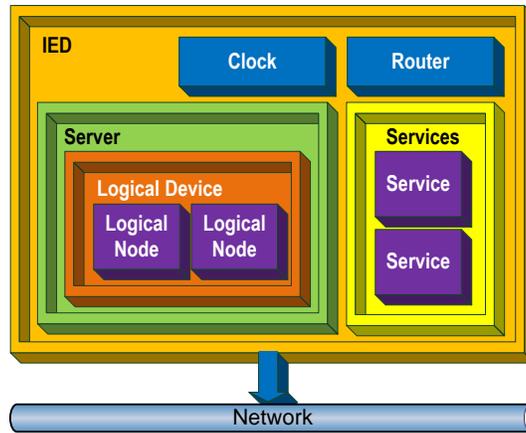


Figure 6 – Typical IED Instantiated Objects

Figure 16 on the Annex shows how this layer is related to the upper layer by the association of the LNode class with the corresponding Substation View classes (Substation, Voltage Level, Function, Power Transformer, Transformer Winding, Tap Changer, SubFunction, Connectivity Node and SubEquipment or Phase), in the upper layer, modeling their deployment by logical nodes in specific IEDs. These objects are also implemented in SCL referring to data classes defined in a Type Model, in the third layer of the integrated model, as part of the Strategic View of an SAS.

### Type Model

The Type Model represents an external view concerning substation information and its formatting types as part of a Strategic View of the SAS. It describes in an abstract way the type of each piece of data used in a SAS and declared in SCL. Only two abstract classes are included in the Type Model by IEC 61850: Data and Type. All concrete data types are derived from these classes, related by the UML diagram of Figure 5, composing the Type Model package, on Layer 3 of the leveled model of Figure 16. Specific types of data for each application are modeled in the next layer.

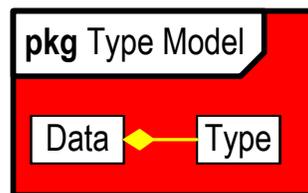


Figure 7 – Type Model Package

Figure 16 on the Annex shows how this layer is related to the upper Layer 2 by the association of the Data class with the LNode class of the Product Model, modeling the structuring of each logical node in Data objects with their Type. The behavior of these objects is dependent on classes from the Communication Model, in the fourth layer of the integrated model, as part of a Strategic View.

### Communication Model

The Communication Model is used to specify the characteristics of local networks and access points, completing the Strategic View of the SAS. It includes the direct communication possibilities between logical nodes by means of logical buses (SubNetworks) and IED access points, as well as which logical devices and logical nodes are reachable across an access point. The Communication Model describes also which IED access points are connected to each sub network. The following abstract classes are included in the Communication Model of IEC 61850: SubNetwork, PhysConn, Address, GSV and SMV. These classes are related by the UML class diagram in Figure 8, composing the Communication Model package, on Layer 4 of the general UML layered model of Figure 16.

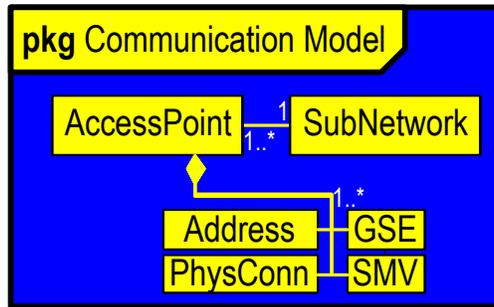


Figure 8 – Communication Model Package

In this model, a Subnetwork object is a connecting node for direct (link layer) communication between access points. It contains all access points which can (logically) communicate with the SubNetwork protocol and without an intervening router. It defines a logical connection with a certain protocol. An Access point is a communication access point of the logical device(s) of an IED to a sub network. A Router class in the Product Model extends access to servers connected to another sub network at another access point of that IED, which hosts the router function, while a Clock class in the second layer synchronizes the internal clocks of all (other) IEDs connected to this sub network. A PhysConn class models the physical address of the server, while an Address class defines the logical address of the server. GSVs and SMVs are classes that define the addresses of Goose and GSSE control blocks, and of Sampled Value control blocks, respectively. Figure 9 shows a pictorial composition of a typical Communication Model using instantiated objects.

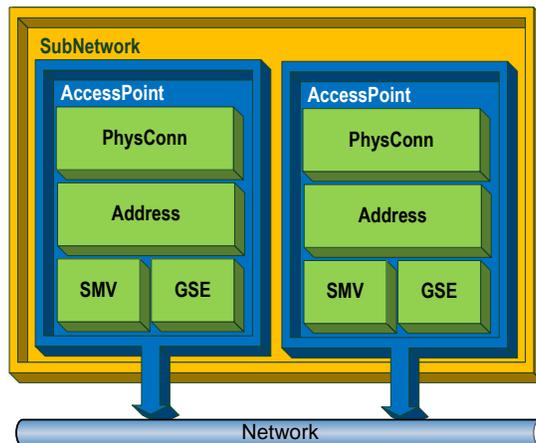


Figure 9 – Typical SubNetwork Instantiated Objects

Figure 16 on the Annex shows how this model is related to the upper layers by the association of the AccessPoint class with the classes LNode e Router of the Product Model, modeling the access

points of logical nodes and routers of each IED. These classes and the remaining objects of the Strategic View are all described in SCL implemented using classes defined in the fifth layer of the integrated model, as part of the Functional View of an SAS.

### Functional Model

A Functional Model defines the task(s) performed by the substation automation system i.e. by application functions, or the characteristic of a system that guarantees the attainment of a specific objective, as part of a Functional View of the SAS. Generally, functions exchange data with other functions that are performed by IEDs (physical devices). Functions may be split in parts residing in different IEDs but communicating with each other forming a distributed function, and with parts of other functions. These communicating function parts are modeled by Logical Nodes, the smallest part of a function that exchanges data, a software object defined by its data and methods. The following abstract classes are included in the Functional Model of IEC 61850: Substation, Automation System, Function, Network, IED, Server, Proxy Gateway, Logical Device, LDO, Compatible Logical Node and PICOM. These classes are related by the UML class diagram of Figure 10, forming the Functional Model on Layer 5 of the UML layered model of Figure 16.

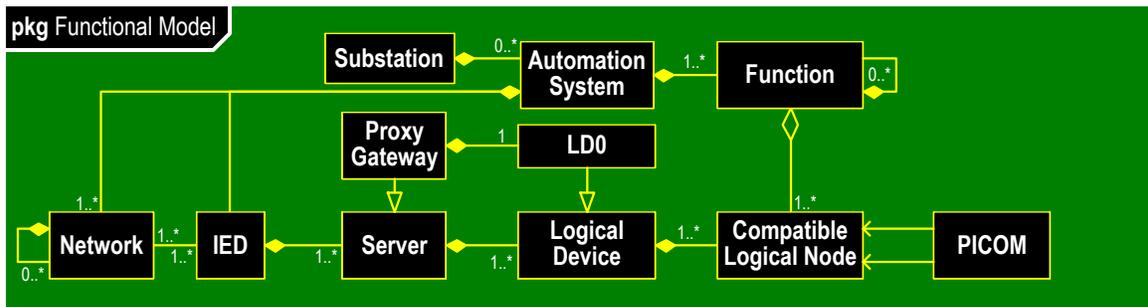


Figure 10 – Functional Model Package

Instantiation from these abstract classes allows the construction of a functional view of an SAS, composed of IEDs that contain Servers with one or more Logical Devices. Each Logical Device may contain several instances of Logical Nodes as containers of Data objects. Figure 11 shows a pictorial composition of a typical functional model with these instantiated objects.

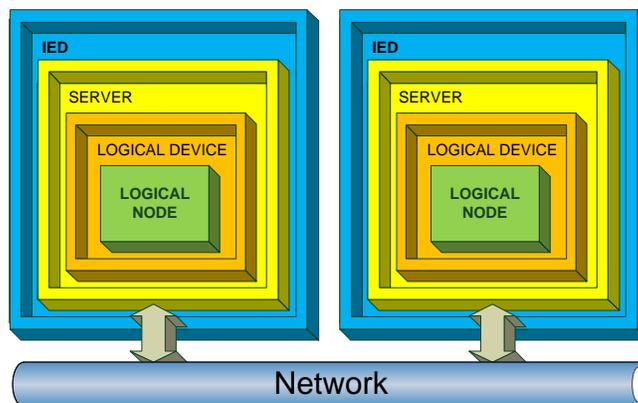


Figure 11 – Typical Functional Instantiated Objects

Figure 12 shows an example of a functional model for a protection IED, constructed from objects inherited from classes of the Functional Model, including logical devices for distance protection, overcurrent protection, measuring, recorder and merging unit.

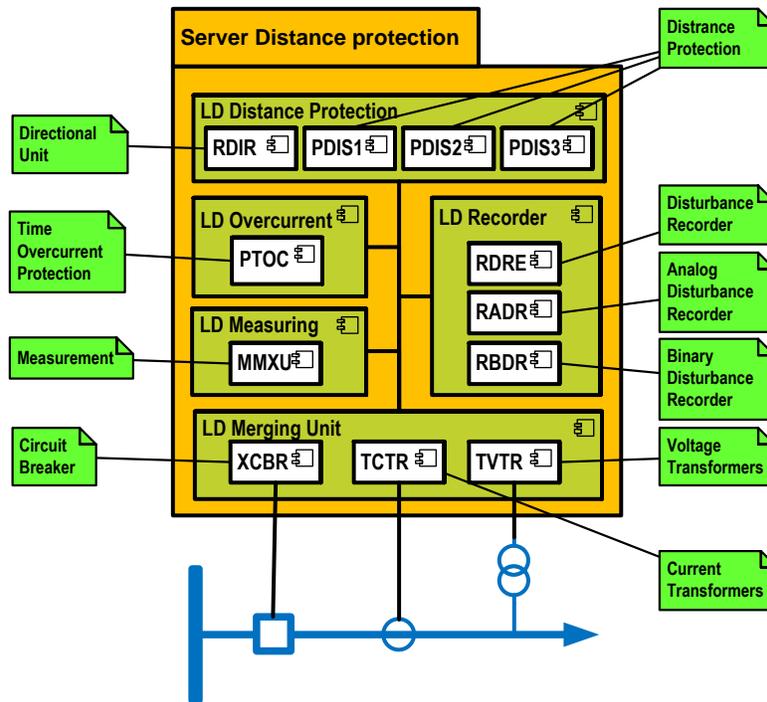


Figure 12 – Functional Model for Distance Protection

Figure 16 on the Annex shows how this layer is related to the upper layers by the association of the classes Substation, Function, Network, IED, Server, Logical Devices, Compatible Logical Node and AccessPoint with the corresponding classes in the upper layers, modeling the concrete implementation of the SAS described by SCL on the upper models. These objects are implemented using data classes defined in a Data Model, in the sixth layer of the integrated model, as part of the Structural View of an SAS.

### Data Model

A Data Model represents the concrete knowledge concerning substation information and its representation in classes, forming a Structural View of an SAS data. It describes in an abstract way the information used by any real function or device in the substation. The following abstract classes are included in the Data Model of IEC 61850: Namespace, Data, Composite CDC, Common Data, Simple CDC, Primitive Component, Composite Component, Basic Type, Data Type, Data Attribute and Functional Constraint. These classes are related by the UML class diagram of Figure 13, composing the Data Model package, on Layer 6 of the UML layered model. The comment boxes in this figure show some derived data objects instantiated from these classes.

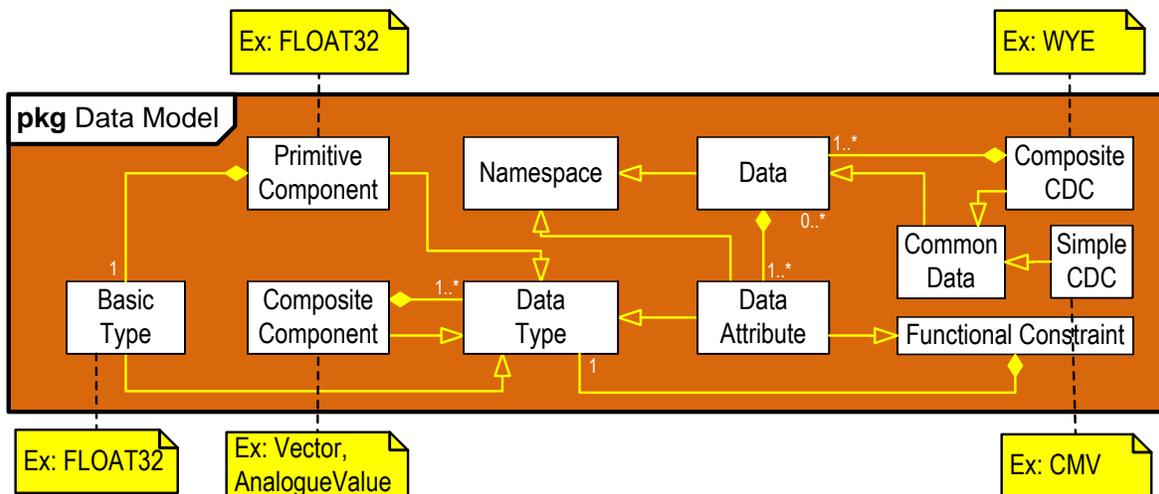


Figure 13 – Data Model Package

A key element of the Data Model of IEC 61850 is the Namespace, an abstract container environment created to hold a logical grouping of unique identifiers (i.e., names), providing context and disambiguation of items with the same name residing in different namespaces. An identifier defined in a namespace is associated with that namespace so that the same identifier can be independently defined in multiple namespaces with different meanings. Figure 16 on the Annex shows how this layer is related to the upper layers by the association of classes, and how a Namespace is inherited by the Logical Device and all Compatible Logical Nodes of the Functional Layer on the upper level, and by containment relations between this with the Data class. The methods supported by these objects are implemented using data classes defined in a Service Model, in the seventh and lowest layer of the integrated model of Figure 16, as part of the Process View of an SAS.

### Service Model

A Service Model is a set of functions and procedures made available for the user of a Data Model, to access or change its data components and configurations as part of a Process View of the SAS. IEC 61850 defines two types of services: Abstract Services – functions and procedures defined for the classes of a model, and Concrete Services – functions and procedures implemented into objects derived from the classes of the model. These services cover functions for read/write values and attributes, control devices, report/log events, access directory information, file upload/download, and transfer of GOOSE/SV messages. An Abstract Communication Service Interface (ACSI) and a Specific Communication Service Mapping (SCSM) provide the interfaces to these services, respectively. The following abstract classes are defined in this model by IEC 61850: Two-Part Application Association, Multi-Part Application Association, Log Control Block, Buffered Report Control Block, Unbuffered Report Control Block, Common Logical Node, Logical Node, Substitution, Control, Data Set, LPHD, LLN0, LOG, File, Time Master, Setting Group Control Block. Additionally, the following Messages classes are defined in the Messages sub package: GSSE Message, GOOSE Message, Unicast Sampled Value Control Block, Multicast Sampled Value Control Block, GSSE Control Block, GOOSE Control Block and SV Message. These classes are related by the UML class diagram of Figure 14, composing the Service Model package, on Layer 7 of the UML layered model of Figure 16. The Messages classes are grouped as a sub package of

this layer, including the GSSE/GOOSE Control Blocks and Messages, SV Message, and Unicast/Multicast Sampled Value Control Blocks, depending on the network middleware.

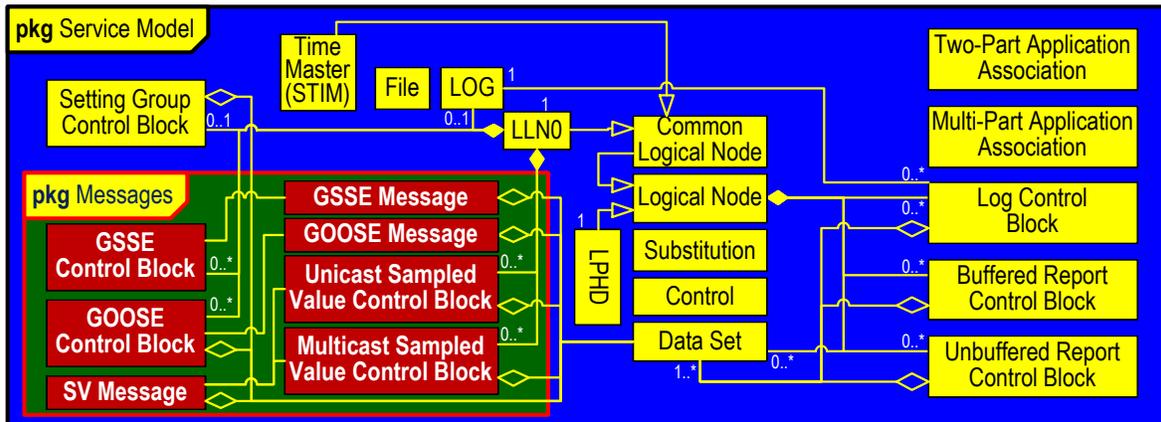


Figure 14 – Service Model Package

Abstract services include Client-server request/reply of information, peer-to-peer real-time multicasting, and time synchronization using SNTP. Figure 15 shows how these services operate as a middleware to the network access to data on the distributed logical nodes and on the mapped physical equipment of substation and power plants.

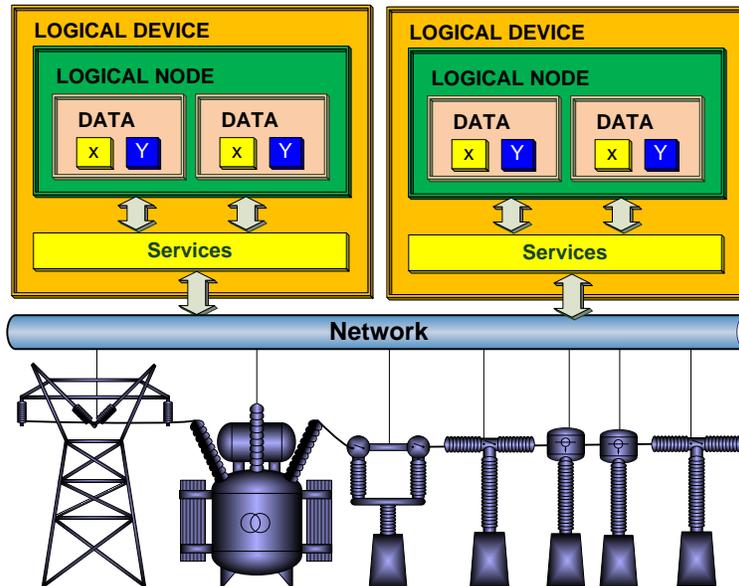


Figure 15 – Typical Service Instantiated Objects

Logical Nodes and their services have been defined to almost any conceivable power system equipment and function, in substations, hydropower plants, wind power and distributed energy resources. Many others are being currently defined. Services have been mapped to the Ethernet protocol stack, Web Services, CORBA (Common Object Request Broker Architecture), and other communication middleware. Figure 16 on the Annex shows how this layer is related to the upper layers by the association of classes to support the composition of messages with data contained in the Data and Functional Packages. These LLmessages are routed directly to the middleware used for communication in the SAS network.

## Conclusion

This paper has shown a top-down object-oriented model of the standard IEC 61850, similar to the traditional seven layers of the OSI model for network communication. All major software classes defined by IEC 61850 are grouped in stacked layers, ranging from the higher slices seen from users and integrators, to the lower layers facing network communication middleware and field process equipment. These seven models compose four concurrent views of an SAS, the Strategic, Functional, Structural and Process View, collected in a unique UML class diagram, covering all major stakeholder needs and classes defined by IEC 61850, suitable as an overview and unified guide to the standard. This integrated model has been successfully used by the author in consulting and in-company workshops and university teaching in post-graduation courses about IEC 61850.

## References

- [1] CIGRE WG B5.32, *Functional Testing of IEC 61850-Based Systems*, Cigré Technical Brochure 401, ISBN: 978-2-85873-088-9, December 2008.
- [2] SIQUEIRA, I. P., *Model Based Testing and Diagnosis of IEC 61850 Systems*, PAC World Magazine, December 2010 Issue, available at PAC World website [http://www.pacw.org/issue/december\\_2010\\_issue/model\\_based\\_testing/model\\_based\\_testing\\_and\\_diagnosis\\_of\\_iec\\_61850\\_systems.html](http://www.pacw.org/issue/december_2010_issue/model_based_testing/model_based_testing_and_diagnosis_of_iec_61850_systems.html).
- [3] IEC 61850: *Communication networks and systems in substations*, International Electrotechnical Commission, Switzerland, 2004.
- [4] OMG, *Unified Modeling Language*, The Object Management Group, available on [www.omg.org](http://www.omg.org), 2005.
- [5] W3C, *Extensible Markup Language (XML)*, The World Wide Web Consortium, available on [www.w3.org](http://www.w3.org), 2006.

**Iony Patriota de Siqueira** is D.Sc. in Electrical Engineering, M.Sc. (Honors) in Operations Research, and MBA in Information Systems. He is a member of CIGRÉ, IEEE and IEC (TC 57 and WG 10), former convenor of CIGRÉ WG B5.32 on Functional Testing of IEC 61850 Based Systems, Secretary of CIGRÉ Study Committee B5, and Chairman of Brazilian Technical Committee of IEC TC 57. Currently his is Vice-Director and Adviser for the Brazilian Maintenance Association, and Manager of Protection and Automation at CHESF. With two published books, in 2009 he received the CIGRÉ Technical Committee Award for contributions to the Protection and Automation area.



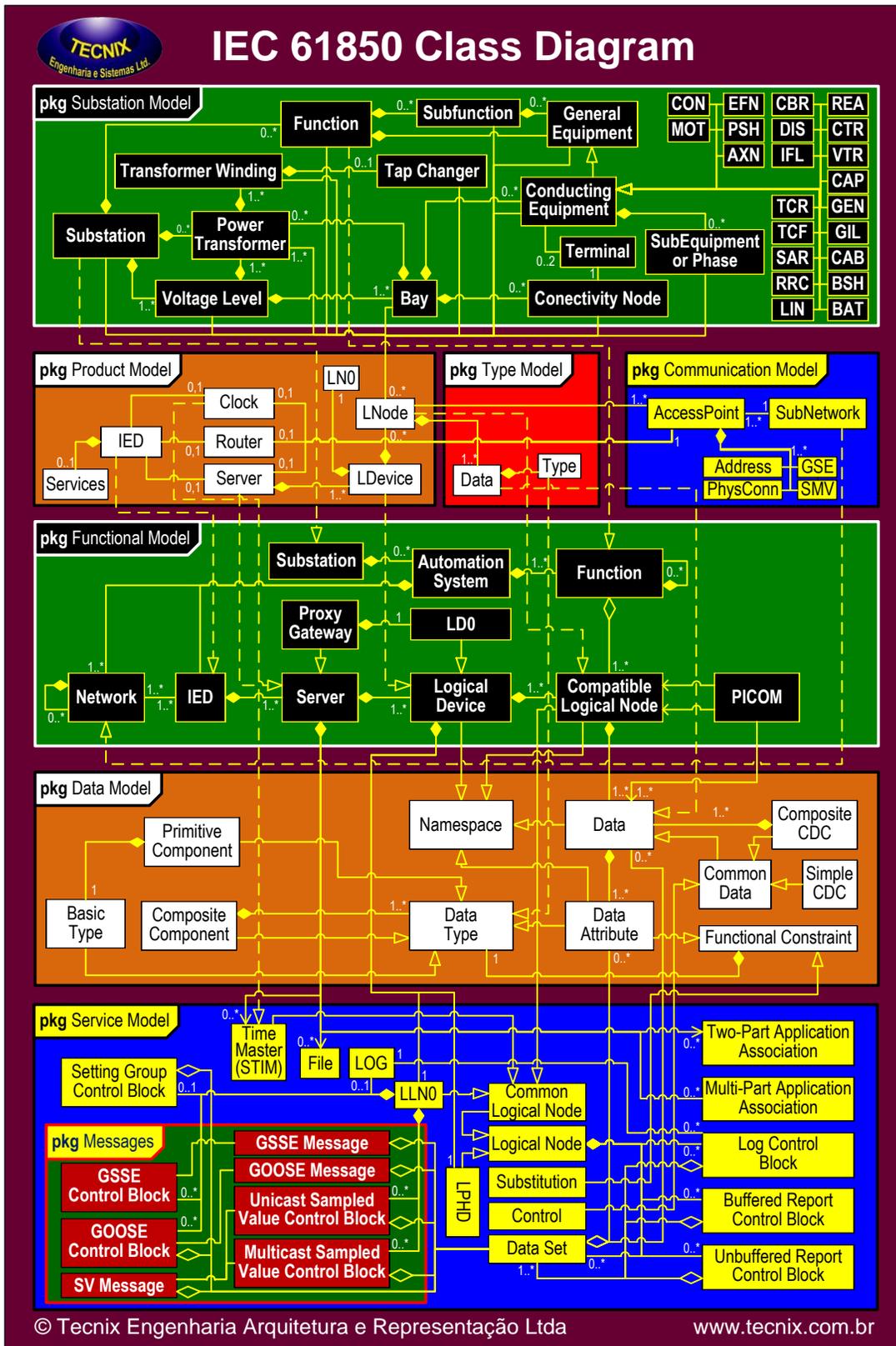


Figure 16 – Layered Hierarchical Object-Oriented View of IEC 61850