IEC 61850 Data Modelling for Utilities

Jörg Reuter, Helinks LLC, Giuseppe Rigadello Enyr SRL

jr@helinks.com

Switzerland

Abstract

Suppliers of IEC 61850 Devices are used to create Data Models which are embedded in their ICD files. Stack manufacturers provide know-how and tools for this task. More and more Utilities are producing IEC 61850 based specifications for Power Automation Systems. They are using IEC 61850 SCL to specify PAC Systems (SSD files) and to describe company standards as well as formalized requirements specification for IED procurement.

Now these Utilities also face the challenge of creating their own IEC 61850 based Data Models. They need to be able to work with NSD files and to build their own Data Model (Data Type Templates) following the IEC 61850 design rules. We are presenting the major steps and pitfalls of this process and relevant tool support especially designed for this task.

1 Overview

First, we introduce the concepts of the IEC 61850 hierarchical data model. This will show that, compared to flat signal lists used for classical communication protocols like IEC 60870-5-101 or Modbus, understanding and working with an IEC61850 data model is a certain challenge. We will explain why a Utility should take up this challenge and invest in some serious IEC 61850 modeling activities.

Naturally not only Utilities are facing this challenge. Every vendor who offers IEC 61850 communication products as well as standardization bodies which want to extend the IEC 61850 Model to further application domains need to dig deep into IEC 61850 modeling.

The IEC 61950 working Group (TC57/WG10) has recognized the need of supporting these activities and has issued a document type specifically addressing the rules on how to define and maintain IEC 61850 Data Models.

We then want to show how to apply these rules and what Utilities must do to be successful with their modeling approach.

2 What is an IEC 61850 Data Model?

1.1 Syntax, Semantics, Encodings and Scope

IEC 61850 is a collection of semantical protocols for different kinds of communication in power systems. The decisive word in this sentence is “semantic”. The data which is communicated is not only based on universal data types like Integer, Boolean etc…, but it has a well-defined, formalized semantic representation. This representation is based on a formal Data Model. This model defines

- Syntax for data
- Mappings and encodings for Data
- Data elements based on specific semantics

That is what we call the IEC 61850 Data Model. This Data Model has various aspects and representations. Different syntax and encodings are depending on the purpose and usage context.

Some examples:

1. ASN1/MMS encoding for online information exchange
2. XML representation for the exchange of engineering data (SCL)
3. Word/PDF tables (e.g. 61850-7-4) or XML (NSD or SCL files) for the sematic definition of Data.

The scope of the Data is not limited to the observable and controllable physical values of a power system, but covers also the description of the power system itself (Primary Equipment, Power Network, Generation Facility, Substation) and the description of the Control and Protection hardware and their network connections (Secondary Equipment).

3 Importance of the SCL Data Type Template Section for the IEC 61850 engineering process

In this paper we are focusing on the aspects of an IEC 61850 data model which are important for Utilities that want to stay in control of the design and engineering of their IEC 61850 based systems. The most important part in this case is the Data Type Template Section of SSD or ISD Files. The Data Type Template Section is important, because it defines the exact set of process data which can be communicated over IEC 61850 in the protection and automation system. It represents the basic definition for protection and automation schemes and for monitoring and control services.

The image below shows four blue quadrants which represent the 4 sections every SCL file has. The left lower quadrant is the Data Type Template Section. It contains the type definitions for Logical Nodes, Data Objects and Data Attributes.

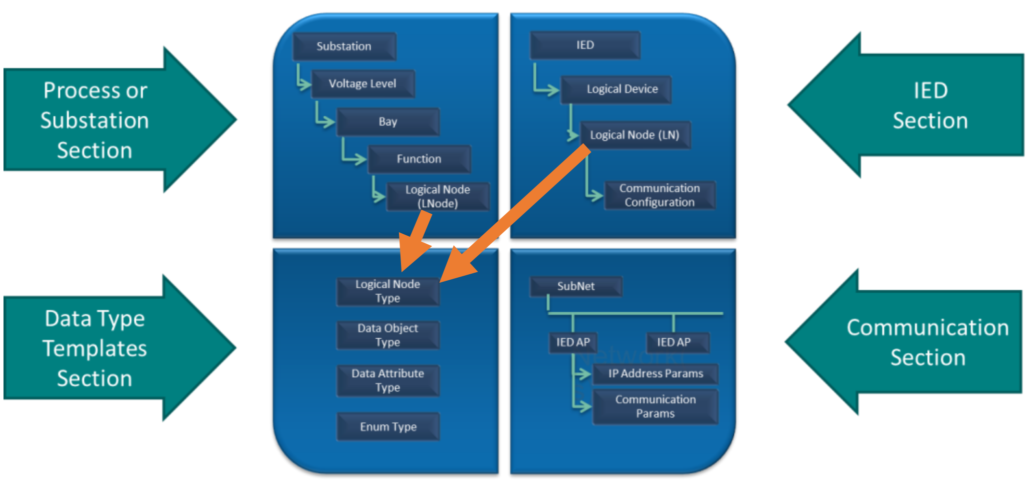

Figure 1. Data Type Templates in SCL

The Logical Node Type defines the Data Objects of a Logical Node. Each Data Object is then defined by a Data Object Type.

The Data Object Type is an assembly of Structured Data Objects or Data Attributes. The set of allowed Structured Objects and Attributes of a Data Object Type is limited by the so-called Common Data Classes (CDC). They are, even if not explicitly defined in the Data Type Template Section of an SCL file, one of the most important parts of the IEC61850 standard. You can find their definition in IEC 61850 Part 7-3.

Like Data Objects, Structured Data Objects are also defined by Data Object Types, Data Attributes are defined by Data Attribute Types. In the simplest case a Data Attribute Type is a Basic Type like Integer, Floating Point or Boolean. But also, Attributes may be nested or can be complex types like predefined or even customized Enumerations. Figure 2 sketches the inner structure of the Data Type Template section.

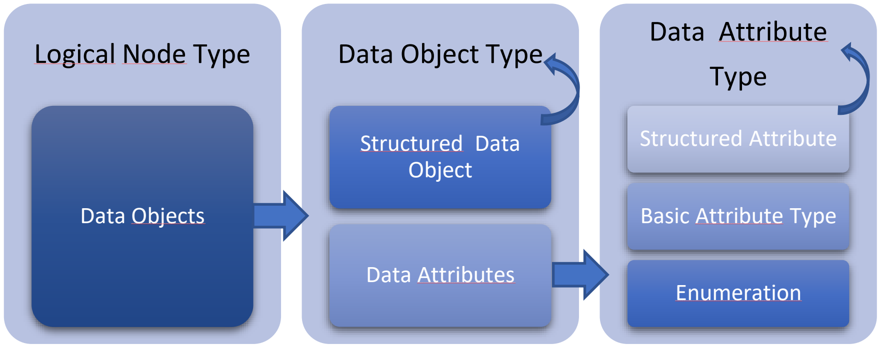

Figure 2. Data Type Template Hierarchy

Based on the Data Type scheme, a data point in a power system can be described in a hierarchical standardized way (Object Reference):

e.G.: MMXU1.A.phA.cVal.mag.f

The Object Reference as shown above is an example of a nested Data Object/Attribute where each name component is based on one of the types defined in the Data Type Template section.

Logical Nodes can now be defined by simply referencing a Logical Node Type. This defines the whole inner structure of the Logical Node.

The Logical Nodes are then grouped into Logical Devices in order to represent the capabilities of an IED (ICD file) or they are grouped into Functions and allocated to the SCL process section in order to specify the PAC System.(SSD file)

4 Why should utilities care about IEC 61850 Data Modeling?

Many utilities used for years now to buy turn-key PAC-Systems. If there is no doubt about the advantage of the IEC 61850 communication protocols in a running system, the engineering process itself and the maintenance of such a system was in the past and is still today a challenge for utilities.

The reason for this is shown based on the following picture:

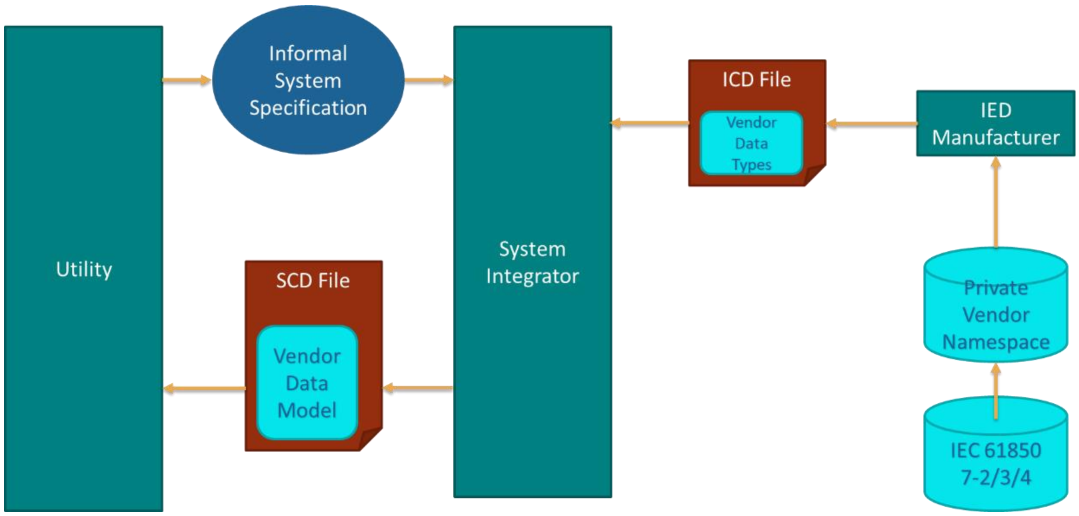

Figure 3. Engineering workflow with Vendor driven Data Model

Utilities used to produce sophisticated, but informal System Specifications for a PAC System. These specifications consist generally of:

1. Single Line drawings
2. IO Lists with different sets of attributes for the signals
3. Naming and design rules
4. Network requirements
5. Application schematics and protection specifications
6. etc.…

The System Integrator has the task to build this system using Control and Protection Relays (IEDs) from different manufacturers. The IED manufactures, mostly by means of the IED configuration tools (ICT), are providing ICD files which describe the capabilities of an IED in terms of a 61950 Data Model. IED manufacturers have build detailed and sophisticated IEC 61850 Data Models which reflect the firmware implementation in the IED.

Major steps of the System Integrators work are:

1. Select the IEDs in accordance with Utilities requirements
2. Allocate the Logical Nodes of the IEDs properly to the process
3. Ensure proper signal mapping between the IO lists and the IEC 61850 addresses
4. Properly implement the Control, Automation and Protection schemes

Now it is obvious that, whenever the detailed design decisions are made by the System Integrator, there will be always different designs even for similar systems while there are different teams at work.

The fact, that different IEC 61850 data models from different vendors are integrated, is also not extremely helpful for building standardized solutions.

Sometimes interoperability issues prevented building a single, consistent SCD file or the SCD file is incomplete because it contains only publishing, but no subscription for GOOSE.

Utilities made the painful experience that these systems are hard to understand and even harder to maintain. Missing documentation of GOOSE and SV applications made it often impossible to understand and track the application design.

Many Utilities currently are looking to improve this situation. The way to do this is leading them to think about their own Private IEC 61850 Data Model, that exactly reflects their needs and that can be used to impose design, data point mapping and the IED structure to System Integrators and IED suppliers.

Based on a Private IEC 61850 Data Model, the workflow in the engineering process changes:

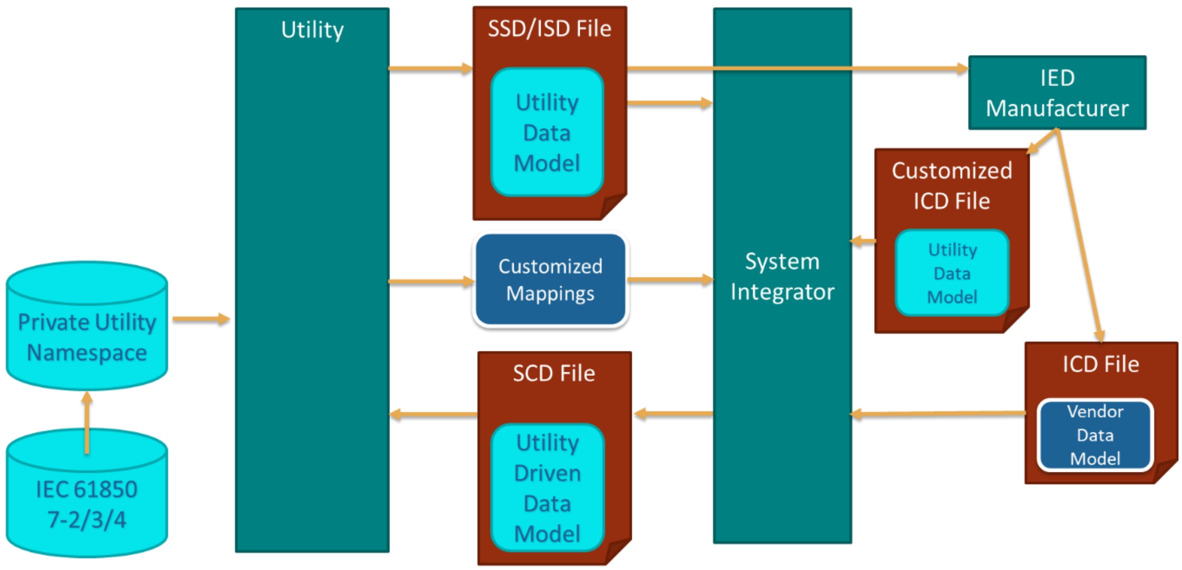

Figure 4. Engineering workflow with Utility driven Data Model

Instead of an informal System Specification the Utility now can build a formalized system specification and deliver SSD and ISD files as core part of a specification. SSD and ISD files are XML files. SSD stands of System Specification Description and ISD for IED Specification Description.

Please observe that an SSD file also may contain IED specifications (Virtual IED).

These files have a Data Type Template section that is based on the Private IEC 61850 Data Model of the Utility.

With the most recent generation of IEDs IED Manufacturers are now supporting a flexible Data Model. In the context of a PAC project the System Engineer can customize the IED Data Model. If an SCL based IED specification is present (ISD file or integrated in the SSD file) the design of the customized IED is not generated by the System Integrator or by the vendor, but it is imposed by the Utility and driven by the underlying Private IEC 61850 Data Model.

It is obvious that Systems designed like this are more standardized and much easier to understand and maintain.

But we may also have in a project some “legacy IEDs” where the Data Model is fixed. This complicates things a little bit, but there is also a way to ensure a Utility driven design. Now, that the Utility has its own IEC 61850 Data Model it can define a mapping between this model and the fixed IED model. The System Integrator uses this mapping to integrate the legacy IEDs. Again, the Utility is in the driver seat and is taking the design decisions.

5 How to create a Private IEC61850 Data Model.

1.2 IEC 61850 Namespaces

The first IEC 61850 users that needed to build Private IEC 61850 Data Models where the IED Manufacturers. From the large set of Logical Nodes and CDCs, they had to select the subset of objects to cover the capabilities of their IED firmware implementation. For quite a while this was a manual process based on the IEC paper documents because these building blocks for the Data Model were contained in printed tables in the part 7-1, 7-3 and 7-4 of the Standard.

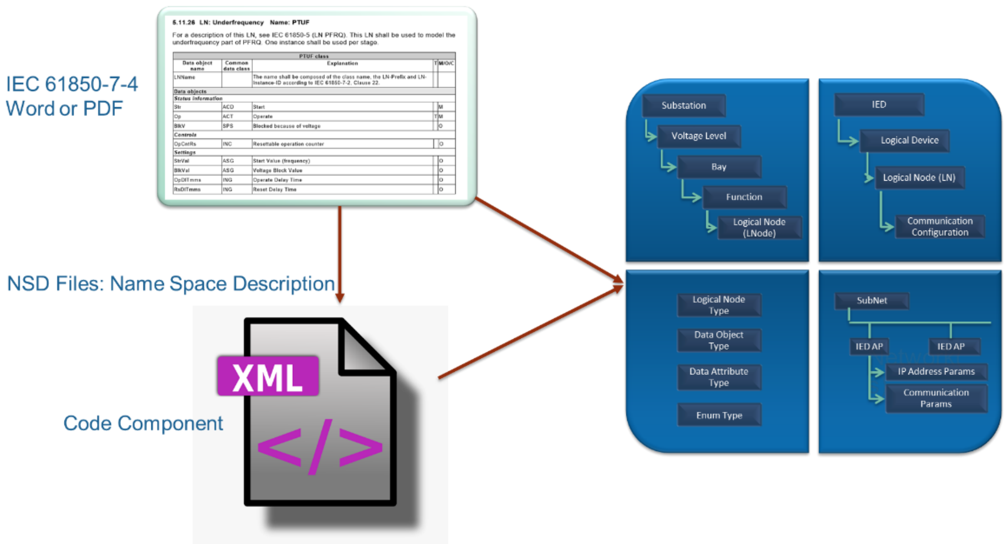

Figure 5. From printed Tables to NSD files.

This manual process was error prone and led to many interoperability issues on run-time and tool level. That is why TC57/WG, the IEC 61850 Standardization body decided to define a new XML scheme and to capture the basic building blocks of a Data Model in a formalized way.

The files are called NSD files (Namespace Specification Description). The collection of Logical Nodes, potential CDCs, Data Objects and Data Attributes and supporting definitions are called an IEC 61850 Namespace.

The document:

IEC DTS 61850-1-2 © IEC 2019

Introduces the Namespace definition, the NSD files and the rules how to build IEC 61850 Namespaces.

1.3 Namespace Types

In this document we also can find the definition of different Namespace types:

- **IEC 61850 Basic Namespaces**

contain the basic classes from which all other namespaces are built, e.g. IEC 61850-7-2; IEC 61850-7-3; IEC 61850-7-4.

- **IEC 61850 Domain Namespaces**

Namespace which extends IEC 61850 covering a specific application domain. E.g. parts 7-4xx

- **IEC 61850 Product Namespace**

Namespace focussing on Products like Switchgear or IEDs and adding specific information about the product via detailed nameplate information.

- **IEC 61850 Private Namespace**

private data model namespace which extends IEC 61850 with the purpose of better supporting a private implementation(IED vendor) or specification (Utility) of IEC 61850.

- **IEC 61850 transitional namespace**

data model namespace which has the goal to expose a proposed extension of the data model to the market, before being formally embedded into a standard (product, domain or basic) namespace

Everything discussed in this papers is related to the Namespace type Private Namespace. If such a Namespace is defined by a Utility or by a IED vendor it needs to be identified and maintained. Namespaces itself have some important properties which facilitates their management and maintenance:

Ownership

A Namespace has an owner who is responsible for the definition and maintenance of the Namespace

Identification

Namespaces are identified by a Name,Version Release and the dependency to other Namespaces

Traceability

Objects can indicate on which Namespace they are based on.

Inheritance

Namespaces can be chained together

1.4 Rules for the definition of Namespaces

Strict rules for the definition and extension of Namespaces prevent owners from jeopardizing interoperability. The rules are different for different types of Namespaces. The rules for Private Namespaces are summarized in the table below:

Operation	Rule for Private Namespace Definition
Data Model	
Extending existing LNs with new DOs	Allowed following DO naming rules within the scope of the product functions
Create new LNs	Allowed following LN/DO naming rules within the scope of the product functions
Deprecate existing LNs	Allowed only by the owner of the namespace
Extending existing DOs enumeration	Allowed under specific conditions and a specific process
Extending CDC with new attributes (DA)	Forbidden
Create a new CDC	Forbidden
Extending existing DAs enumeration	Forbidden
Making more stringent the DO presence conditions on existing LNs	This DO should be tagged as "namespace specific"
Making less stringent the DO presence conditions on existing LNs	Forbidden
Deprecating DOs from existing LNs	Allowed
Making more stringent the DA presence conditions on existing LNs	In some very specific cases
Making less stringent the DA presence conditions	Forbidden
Defining rules for LN prefix/suffix	Forbidden
Defining rules for LD arrangements	Forbidden
Communication services	
Extending the ACSI	Forbidden
Providing a new SCSM	Forbidden
SCL language	
Extending the SCL grammar (XSD) with SCL new tags and new attributes	Forbidden
Extending the SCL grammar (XSD) through private sections	Allowed
Extending the SCL grammar (XSD) with tags and new attributes from other xml namespace	Allowed

Following these rules ensures interoperability and makes it possible for IED vendors to match their implementation with a Private Namespace of utility.

Please observe that rules for Logical Node Prefix/Instance and also the specification of Logical Device arrangements are not allowed in the Namespace definition. This does not mean a Utility cannot define rules for Prefix/Instance naming and Logical Device arrangements, but they may change depending on the kind of project or application. These rules must not be enforced by the Namespace definition. However, they can be applied when instances of Logical Nodes are created based on the Namespace.

6 How can a utility build a Private IEC 61850 Namespace?

The first step is to get the Management awareness and commitment for this activity. If the utility recognizes the benefits in terms of project and maintenance costs and sustainability, an important step is done.

The second step consist of building up IEC 61850 competence. It is important to define one or several a Data Model Manager in the organization. These Data Model Managers are responsible to build the Private IEC 61850 Namespace.

This work could be done in Excel, but given the fact that NSD and SCL files are XML based and follow strict rules, it is recommended to use at least generic XML tools. It is even better to rely on specific IEC 61850 tools which are supporting the IEC 61850 Data Modeling work.

To create a comprehensive set of standardizes artifacts for the IEC 61850 Uitility Data Model the Data Model Manager has a lot of work to do:

- Start with NSD files
- Build Data Type Templates
- Build Function Templates
- Build Bay Templates
- Build Application Templates (BAP)
- Build ISD and SSD Files for projects based on the templates

If you look at the list above it is obvious that the Utility Namespace alone is not covering everything needed for a formalized specification, but the Namespace represents the foundation on which the more advanced artifacts like Function, Bay and Application templates are build on.

Finally, the Utility Data Model must be embedded in the companies processes and organization. This requires training and rethinking the interface between departments and even between customer and supplier.

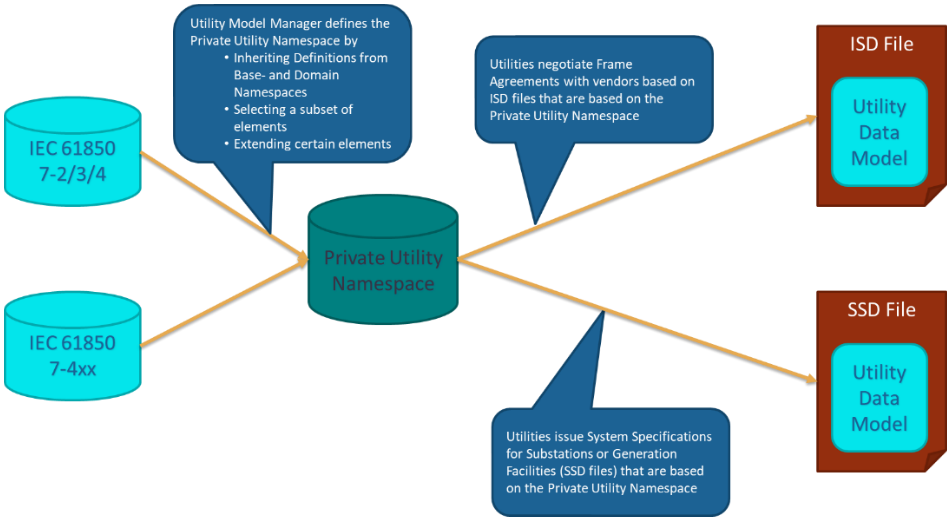

Figure 6. Utilities standardization process .

7 Conclusion

By creating their own IEC 61850 Data Model

- Utilities stay in control of the design of their applications and systems.
- They get similar solutions and designs for similar Substations/Generation Facilities.
- Maintenance becomes feasible with reasonable effort.
- PAC projects can be run more efficiently.

But, there is no “Free Lunch”.

- Building up IEC 61850 knowledge requires an effort.
- IEC 61850 has a steep learning curve It needs training and hands-on activities. (lab environment, pilot projects etc.…)
- Do not shy away from using consultants for initial support

Jörg Reuter, Helinks LLC

Giuseppe Rigadello, Enyr SRL