HOW IEC 61850 HELPS DURING DEPLOYMENT AND OPERATION

Vinoo S Warrier, Kalkitech, +91 95357 82230, vinoo@kalkitech.in
Sterin T Jose, ASE, a Kalkitech company, +1 925 663 9833, sterin.jose@ase-systems.com

Introduction

IEC 61850 has been improving the process of substation engineering and commissioning from the time of its introduction. Ed.2.0 improvements cleared the confusion on signal mapping configuration of reports, and GOOSE and Sampled Values between IEDs. Definitions and identification of client systems and their data acquisition capabilities are represented in a manner that is user friendly and easy to understand. Routable GOOSE and Sampled Values expanded the boundaries of signal exchange to allow inter-IED signalling across substations. Security concepts are also getting implemented in parallel to make sure that such data exchanges are not subject to eavesdropping and attacks. This marks the success of IEC 61850 in becoming a truly globally accepted standard. This paper examines the evolution of IEC 61850 from its advent until now and illustrates how its evolution has been driven by feedback from field experience gained from deployments.

As the industry moves forward quickly with IEC 61850, this paper walks through real life use cases and situations which utilities often encounter during typical substation engineering, commissioning and operations. The recommended IEC 61850 engineering process starts from SLD definition, assignment of IEDs into bays, configuration of signal mapping between IEDs to realize breaker control use cases, simulation of protection trigger conditions using test kits and finally verifying the IED actions through breaker simulators - all using the framework of IEC 61850 tools, network and IEDs. This paper also covers scenarios where this standard approach is not followed, and how IEC 61850 can help to mitigate configuration issues under such conditions.

An impressive aspect of the IEC 61850 is the change management process that can be handled at different stages by different stakeholders to achieve a collective goal. There will be ambiguities during the execution, however, the standard helps to handle such confusion effectively to meet the final objective of the communication system. This paper offers a brief overview of the process and advantages from engineering and test perspectives, which mostly occur during deployment and commissioning of the system. It includes a discussion on how to ensure interoperability between products that comprise IEC 61850 deployments.

Evolution of IEC 61850

The evolution of IEC 61850 has been a continuous process since the first edition was published in 2004 from the stables of the WG10 under the IEC Technical Committee TC57. No discussion on IEC 61850 is complete without mention of its acclaimed system engineering process and standards rooted around its own configuration language called SCL. It has been one of the first and possibly the only communication standard which has ventured beyond being just a protocol but instead being a complete substation engineering and operation standard. During this period we have seen the first implementations of IEC 61850 launched by nearly all OEMs along with the necessary tools for configuration of the substation system, make their appearance in pilots, in test beds and then in production environments around the world. We have seen in addition to the OEM offerings, a large ecosystem of tools from third parties evolve to support the growing requirements and expectations as a natural and desired consequence of creating and utilizing an open standard. We have seen the evolution of tools for substation specification and configuration, IED configuration and tools for protocol testing and conformance testing as well as a conformance test facility accreditation program with worldwide acceptance. The industry has benefited from a large contingent of experts and consultants and training/enabling programs, the open support forum UCAIUG, as well as conferences and workshops that have grown around the world. The latest in this series of developments has been the advent of IEC 61850's own interoperability live test lab, the UCAIUG IEC 61850 IOP event which was successfully conducted for the third time in October 2017.

On the functional front, IEC 61850 has grown to cover new scope and areas of operation beyond just being a station bus protocol, making its presence felt in the process bus, embracing models for newer domains like hydro, solar and distributed energy resources, and providing for an alternative to the popular C37.118 synchrophasor protocol for Wide Area Monitoring System (WAMS). It has also expanded its geographic limitations beyond the substation boundaries and is now actively developing standards that allow it to be utilized between substations and substations to control centers (as a possible replacement for conventional SCADA-RTU protocols)

On the technology front, we are already seeing exercise of the "abstract" nature of the IEC 61850 standards which was designed to allow it to operate on different and newer carrier technologies that naturally evolve with time. The first such exercise is the design of a new SCSM, the IEC 61850-8-2 standard, that maps the ACSI to an XML based carrier, a work in progress at IEC. This provides a radically different technological alternative to MMS which has been the default carrier in the first SCSM IEC 61850-8-1 for IEC 61850 client server messaging. MMS has drawn some criticism, despite being a ideal match for IEC 61850 at the time of its advent, mainly around its complexity from an implementation perspective. The arrival of IEC 61850-8-2 is a welcome change at this juncture that reinforces the flexibility of the standard and more alternatives may be expected.

Being a pioneer in any domain is always fraught with issues and IEC 61850's engineering methodology is no exception. The SCL based engineering system has been plagued with interoperability errors springing around the smallest loophole or ambiguity in the standard. The communication between IEDs and clients as well as inter-IED communications that utilize the "protocol" part of the standard has stabilized rapidly and exhibited sufficient interoperability in the early stages. This was to be expected since "protocol" testing and alignment with standards has been a standard practice and is not an alien concept to the power systems industry. However the configuration subsystem and SCL engineering method appeared to take a little longer to stabilize. This aspect was finally speeded up with significant contribution from OEM participation in the standard bodies as well as feedback from the IOP plugfest events.

While the changes described above provide a macro picture of the evolution and growth of IEC 61850, it would be beneficial to examine specific changes that exemplify how IEC 61850 has improved the deployment and operations experience in substation automation projects. The next few sections describe this in detail.

IEC 61850 Engineering Process

The engineering process defined in IEC 61850 serves as an excellent example of the responsiveness of the standard to the requirements of the industry based on their experience in deployments the field. The standard engineering philosophy in IEC 61850 originally started with an idealistic "top-down" approach with a sequence starting with the creation of a substation specification, examining the capabilities of IED offerings and matching them to the specification requirements and then mapping those functions to generate a configured substation description.

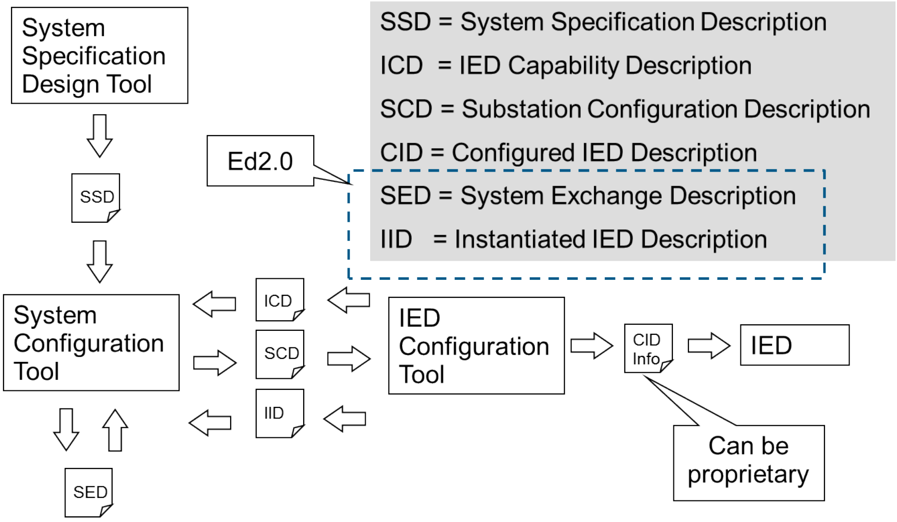

System Specification Design Tool
SSD
Ed2.0
SSD = System Specification Description
ICD = IED Capability Description
SCD = Substation Configuration Description
CID = Configured IED Description
SED = System Exchange Description
IID = Instantiated IED Description
System Configuration Tool
ICD
SCD
IID
IED Configuration Tool
CID Info
IED
SED
Can be proprietary

Fig. 1: Standard Engineering Philosophy in IEC 61850

Originally the process considered a completely top-down approach as described above. This was in conjunction to the waterfall model employed in the software development life cycle and hence suffered from the same drawbacks associated with that model – namely the inability to address real-life changes that occur sometimes in the field or when configuring the actual device. It is not always practical to bring those changes back as a specification document and start the top-down approach from the SCT again to handle the modifications. Consider as an example a case where at the time of configuring the IED from the SCD, the need for an additional Data Object in the LN Model is recognized. This is a local matter and can be easily executed on the ICT, however editing the model only locally would cause cascading errors for all other IEDs or clients that receive data from that LN. Therefore, it then requires to be brought back as a documentary note to the SCT station to restart the top-down flow again for the benefit of all the other IEDs that are recipients of that data. Exchanging information between tools like the ICT and the SCT using text files, emails or documents is exactly the problem that the Substation Configuration Language (SCL) was intended to eliminate. Therefore Ed2.0 of the standard brought in the possibility to provide a standardized reverse-flow by allowing the ICT to modify required elements of the configuration for a specific device not only in the direction of the device being engineered, but also generating an IID file that can be reverse-imported into the SCT to update the overall system configuration, thereby eliminating non-standard document flows.

The IID file is not entirely designed for the problem statement defined above but in reality is a vehicle for many other use cases which also require a configuration information update back to the SCD. A field replacement of a faulty device or card, a field upgrade of a device or application of a security patch and many such use cases may result in a material change in the version numbers or data model of the device that needs to be synchronized back to the SCD. Such updates would normally be performed by the vendor provided ICT and hence it is the ideal tool to regenerate an updated IID that can be fed back into the SCT.

The SED file is another new entrant with Ed2.0 that is designed to allow exchange of a subset (whole or part) of a configured system with another SCT, presumably one managing a different system configuration. This would come into play within use cases where substation to substation communications are being engineered where each substation may have its own SCD to start with, but requires mapping signals from the other.

Thus it may be said that the current state of IEC 61850 engineering processes and tools provide for top-down, bottom-up as well as a mixed-mode of engineering, thereby providing the highest extent of flexibility to meet varying requirements.

Role of SCT and ICT

While the concept of a standardized language for engineering information and workflow between well-defined tools for managing the engineering process was well managed in the first edition of IEC 61850, there were still issues faced in the field on application of these tools. The SCL defines a lot of useful markup to delineate what information is defined where and what information can be modified by what tool, but there were still issues related to the division of activities between the SCT and ICT.

Ed2.0 acknowledged this issue and introduced a section that defines the role of the SCT and ICT clearly. For example, it defined ICT rules related to removing unsupported enumeration values. It also clarifies the role of the ICT in modifying configuration values and setting parameter values when importing an SCD and adds two new version tracking parameters to track the same. These are added in the LLN0.NamPlt DO as new attributes “valRev” and “paramRev” in addition to the pre-existent “configRev” attribute that tracks changes in the overall LD. The SCT is also allowed to modify configuration and setting values and appropriately increments the tracking values.

Further it adds a new conformance statement for tool vendors called SICS (SCL Implementation Conformance Statement) that clearly defines mandatory capabilities for ICTs and SCTs and a defined list of optional capabilities which are to be filled in by providers of these tools.

Testing in Live Systems

The ability to test an automation system or part of an automation system while continuing to run a significant part of it has always been a requirement in the power industry. This was designed for in the first edition of IEC 61850 by providing for multiple modes of operation defined for an entire logical device or even down to specific Logical Nodes. Modes named as “Test” and “Test/Blocked” could be used to temporarily switch the operating mode of an entire LD or of a specific LN to process “test” inputs instead of “real” inputs. The flagging of input data as “test” or “real” was done by a special bit in the incoming supervisory command or GOOSE message.

Consider an example of a use case that uses a breaker controller IED that is operated by an incoming GOOSE message from a protection relay. It is desirable to test the entire use case with test data injected by a test set (instead of the actual trip message from an actual relay) but at the same time preventing the actual operation of the breaker. IEC 61850 edition 1 provided for a controllable "Mod" data object that could be commanded from a client to set any LN (or the entire LD by controlling the "Mod" of its LLN0) to a value like "Test" or "Test/Blocked". Setting the breaker controller IED's LD or just its XCBR LN to "Test/Blocked" was intended to allow operation of the function without actually energizing the breaker control relay. This mode would disallow the receiving IED to operate off a real trip message but allow it to proceed with a "test" trip message (indicated by a "test" bit in the incoming GOOSE).

However, it was not sufficiently clarified as to how the receiving IED could be configured to allow real trip messages from a real IED as well as simulated test messages from a test set at the same time and operate off one of them depending on its own mode. It was also not clarified how the operation of the function could be reported back to the test kit when there is no actual exercise of the breaker.

Ed2.0 provided for a detailed explanation of Mod/Beh handling as an annex in 61850-7-4. An excerpt in Fig-2 below shows the required behavior when "Beh" is "test/blocked" to different kind of inputs. As evident from the table, a supervisory control command from a client software that has the "test-status" service primitive set to true would receive a positive acknowledgement even though there would be no output to the actual switchgear.

MODE/BEHAVIOR	**Test/blocked**
Function behind LN	ON
Output to the Process (Switchgear) via a non IEC 61850 link for example wire (typical for X..., Y..., and GGIO LNs)	NO
Output of FC ST, MX (issued independently from Beh)	value is relevant q=test +operator-Blocked
Response to (Normal) Command from Client (a+ / a-acknowledgement)	a- neg. ack.
Response to TEST Command from Client (a+ / a-acknowledgement)	a+ neg. ack.
Incoming data with q=normal	Processed as valid
Incoming data with q=operator Blocked	Processed as blocked
Incoming data with q=test	Processed as valid
Incoming data with q=test +operator Blocked	Processed as blocked

Fig. 2: Behavior of an LN in the "test/blocked" mode

Ed2.0 also provided for the following changes that enabled more completely the goal of testing a live system.

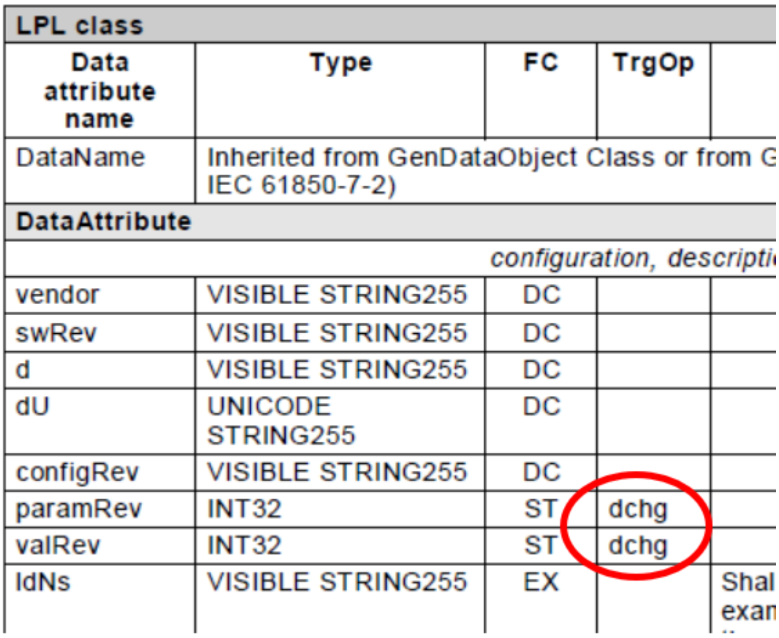
- Any LN can contain an "InRef" data object (inherited from the Common LN class) which contains attributes to set two different kinds of input signal references at the same time, a SrcRef for the actual signal as well as a TstRef for a test signal. Another attribute called "tstEna" in the InRef object allows control of switching the LN to use the TstRef instead of the SrcRef.
- Additionally the CDCs related to the controllable classes also added two attributes "OpRcvd" and "OpOk" which trigger off the receipt and successful acceptance of a control command, regardless of whether the actual physical output was fired or not.

These two sets of changes now provide a mechanism for a test kit to set its own GOOSE message as the TstRef to the LN under test and then set its "tstEna" to true for a short duration of time, sufficient for it to verify that the OpRcvd and OpOk changed state. Following the completion of this test it could easily set the "tstEna" back to false to enable the LN to continue to operate off the real signal.

Version and Change Management

A substation automation project always leads to large volumes of configuration data for each IED, the network and the system as a whole. The version control and change management of this volume of data is a critical aspect that if not properly handled could lead to catastrophic failures. IEC 61850 addressed these requirements with multiple provisions.

- SCL files have a header section with provisions for a history tracking that can accumulate a record of every significant change with a version number. This can be applied to any kind of SCL file including ICD, SSD, SCD etc. but is primarily required in the SCD file since it is the repository of all configuration information of the system as a whole.
- IEC 61850 relies on the structure of Datasets to be known across the clients and servers operating in a configured system such that most transmissions of data only convey the encoded data as a structure and receiving IEDs/ Clients can parse the data out of the incoming packet only because they know the membership, type and sequence of all members of the Dataset. A re-configuration of a Dataset that is performed without synchronizing the change across all IEDs and clients could have disastrous consequences. IEC 61850 provides a "ConfRev" attribute in most control blocks to protect against this scenario. Tools are expected to increment the ConfRev of any control block that references a Dataset that is being changed and receiving devices are expected to check the ConfRev and alarm the user if un-synchronized changes are detected.
- Edition 2 has introduced a mechanism to track online changes in configuration data (initialized attributes of "CF" functional constraint) as well as setting parameters ("SG/SE" parameters or "SP" parameters). This consists of two attributes added in the Logical Node NamPlt of LLN0
  - paramRev – changed when a setting value is changed either through the SG/SE mechanism or through a direct write to SP values
  - valRev – changed when a configuration value ("CF") is changed
- These parameters trigger the "dchg" TrgOp and hence can be configured to be reported or logged on change.



LPL class				
Data attribute name	**Type**	**FC**	**TrgOp**	
DataName	Inherited from GenDataObject Class or from G IEC 61850-7-2)			
DataAttribute				
		configuration, descripti		
vendor	VISIBLE STRING255	DC		
swRev	VISIBLE STRING255	DC		
d	VISIBLE STRING255	DC		
dU	UNICODE STRING255	DC		
configRev	VISIBLE STRING255	DC		
paramRev	INT32	ST	dchg	
valRev	INT32	ST	dchg	
ldNs	VISIBLE STRING255	EX		Shal exan

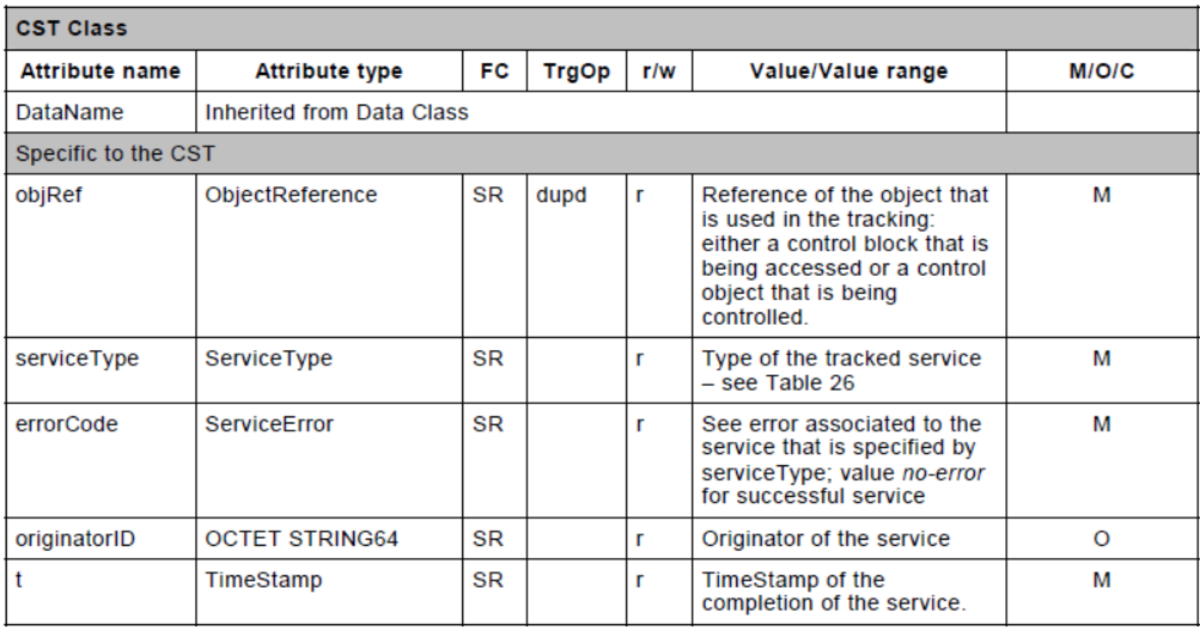
Fig. 3: paramRev and valRev in NamPlt class

Service Tracking and Logging

IEC 61850's Report By Exception mechanism was designed around attributes that raise a TrOp on change where the association of an attribute with a particular TrgOp was designed into the standard and not left to user choice. Control blocks used in reporting and logging would monitor a Dataset for these trigger raises and perform the reporting or logging function which was the basis for a SOE (Sequence of Events) record creation.

However, in the practical case there were still many unsatisfied use cases related to the ability to create and maintain a detailed log that can be used to rebuild a fine grained sequence of events in case of an after-the-fact analysis. For example, a successful control command could be logged by the status change of the "stVal" which raises a "dchg" trigger but what about failed control commands. Additionally, all attributes of all control blocks had no TrgOp attached and hence any parameterization of a control block could not be logged or reported. For example, a user disabling a critical report control block, could not be captured. A post mortem analysis would yield no information of the critical action that was performed. The paramRev and valRev mechanisms listed in the previous section do not apply since these are not CF, SG/SE or SP attributes.

Ed2.0 added a rich suite of LNs, CDCs and service models to enable exactly the above use cases. The root of the tracking solution is a new CDC called “CST” for “Common Service Tracking” which has the following attributes:



CST Class						
Attribute name	**Attribute type**	**FC**	**TrgOp**	**r/w**	**Value/Value range**	**M/O/C**
DataName	Inherited from Data Class					
Specific to the CST						
objRef	ObjectReference	SR	dupd	r	Reference of the object that is used in the tracking: either a control block that is being accessed or a control object that is being controlled.	M
serviceType	ServiceType	SR		r	Type of the tracked service – see Table 26	M
errorCode	ServiceError	SR		r	See error associated to the service that is specified by serviceType; value *no-error* for successful service	M
originatorID	OCTET STRING64	SR		r	Originator of the service	O
t	TimeStamp	SR		r	TimeStamp of the completion of the service.	M

Fig. 4: Common Service Tracking CDC (Note: Partial extract only shown above)

The service type attribute is an enumeration of every defined IEC 61850 service invocation and hence this object can store the salient details of every request from a client. Note that the objRef attribute raises a “dupd” trigger and therefore all the associated attributes can be reported or logged using an FCD on the DO constrained by “SR” functional constraint.

This could be used to track a control command with its target DO, its sending originator, its result whether it is successful or failed with a detailed error code as well as the time of receipt of the service. But even this mechanism had identified gaps in the use case. For example sending a “close” control command to an already closed breaker would fail with a “position-reached” error code but there was no option to actually log the service parameters of the incoming control command to verify that it indeed was a “close” command that had arrived. There was no option to log the type of the command (Direct, SBO, Enhanced Security etc.) or the control number (which if logged could be used to check against the remote client’s own log of commands) or any other service primitive.

This was accommodated by creating a specialization of the CST class called CTS (for Command Tracking Service) which inherited all the parent attributes of CST but was further specialized to log the incoming service primitives of a control command. This would log, for example, the entire “SBOw” that comes with a Select-with-Value command or the entire “Oper” structure that arrived with an operate command.

Another specialization of the CST class was a set of CDCs for tracking control block parameterizations, one type for each “kind” of control block. The BTS CDC for example was designed to track parameterizations of a BRCB and contained attributes that mirror every attribute of the BRCB

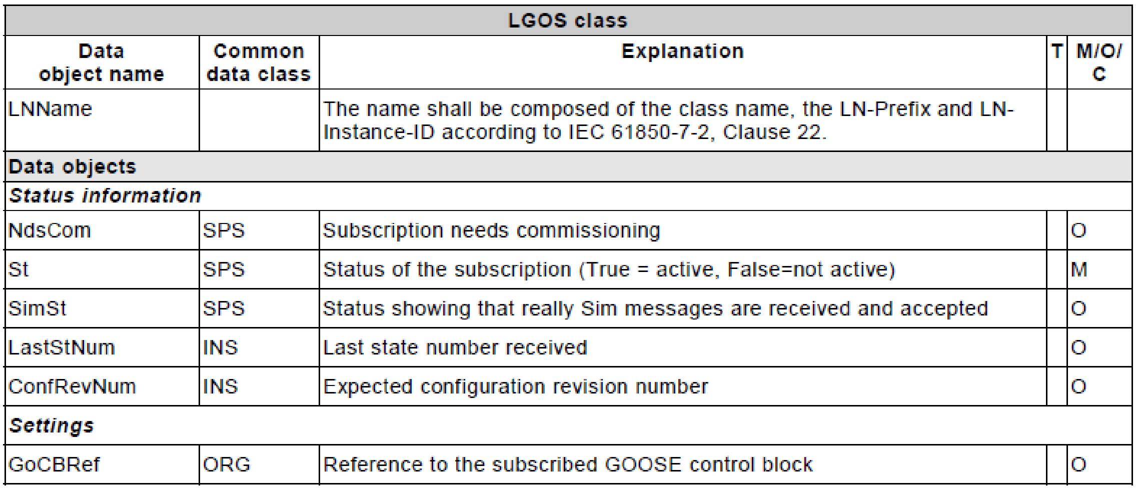
The LTRK Logical Node was defined to act as a top level container for all these tracking objects. One instance of an LTRK contains all the necessary Data Objects that can suffice to track every control command, every control block parametrization and even any other regular service invocation (other than control commands or control block changes)

Supervision of PUB/SUB Subscriptions

The process bus concept was realized in IEC 61850 substations primarily to eliminate the cost and complexity of the cabling that was used to connect the process side equipment with the IEDs in the control panels. This was mainly the wiring from IEDs to the CTs and PTs as well as output signals to the breakers and switches. This cabling was removed by using an Ethernet bus carrying digitized samples of the CT/PT values and GOOSE messages to breaker controllers connected locally to the switch gear. Additionally copper wired interlocks between IEDs were also replaced with GOOSE messages on the station bus.

The primary resistance to moving to the process bus concept was the fear of whether the bus could deliver the performance needed to deliver the samples in real-time, as well as doubts about whether a bus disconnection event would be detected reliably. The publish/subscribe mechanism of GOOSE and SV transfer originally dealt with these issues. A GOOSE message contains a time-allowed-to-live parameter in it which announces to the receiving IED the maximum delay to the next repetition of this GOOSE. Therefore, all receiving IEDs would start a timer when they receive a GOOSE and if the next GOOSE from the same control block fails to arrive before the timer expires this was an indication that the publishing IED had gone down or there was a connection loss. This could be dangerous if the publishing IED is supplying a critical interlock value or performing a critical trip command function. Therefore every subscribing IED is supposed to raise an alarm when they detect a "Publisher down" event. The question was, how? There was just not enough clarity on how this alarm was to be raised and event logged or processed.

Ed2.0 provided the solution to this issue by defining two new Logical Nodes for GOOSE supervision (LGOS) and sampled value supervision (LSVS). Consider the structure of the LGOS LN given below in Fig-5.



LGOS class				
Data object name	**Common data class**	**Explanation**	**T**	**M/O/C**
LNName		The name shall be composed of the class name, the LN-Prefix and LN-Instance-ID according to IEC 61850-7-2, Clause 22.		
Data objects				
Status information				
NdsCom	SPS	Subscription needs commissioning		O
St	SPS	Status of the subscription (True = active, False=not active)		M
SimSt	SPS	Status showing that really Sim messages are received and accepted		O
LastStNum	INS	Last state number received		O
ConfRevNum	INS	Expected configuration revision number		O
Settings				
GoCBRef	ORG	Reference to the subscribed GOOSE control block		O

Fig. 5: LGOS Goose subscription monitoring LN

The "St" and "SimSt" attributes are the operative parts of this new LN. Note that they are SPS data objects and register events on transition between true and false values. "St" tracks the status of the GOOSE subscription and a true-to-false transition event could be processed as an alarm indicating it is no longer receiving timely GOOSE messages.

The "SimSt" is a similar Data Object that tracks the status of simulated GOOSE messages when the receiving IED has its LPHD.Sim set to true.

The LSVS LN is nearly identical in form and function except that it serves to supervise a Sample Values subscription.

Conclusion

This paper examined the evolution of IEC 61850 from the time of its advent as an evolving and interoperable standard, functionally and technologically. It went on to clarify the fact that the first releases while being significantly functional and acting as a game changer in the substation automation industry, still experienced adoption issues and technical or usage related issues in the field. This paper mainly described the feedback and situations from the field and the response from IEC 61850 in addressing those issues in later revisions by incorporating corrections as well as new functions and features. In conclusion, IEC 61850 can safely be said to have evolved as a global standard, maintained its utility in the face of changing needs and evolving technologies and proved itself as a responsive standard adapting to criticism as well as new requirements at an impressive pace.

Biographies

Mr. Vinoo S Warrier heads Kalkitech's product research and development. In this role he oversees product development for the grid communications and automation product family (SYNC), power optimization products (Eltrix), last mile connectivity (SYNCNet) and scalable cloud technology for the back end. He has over 17 years of energy industry experience, designing and implementing communication solutions for power system automation, protocol communication, ICT and Cloud based development. He has actively participated in international standards organizations related to communication protocol standards and has engaged in training and consulting in these fields. Mr. Warrier holds a Bachelor of Technology degree in Production Engineering and Management in from Regional Engineering College, Calicut (now NIT-Calicut), India.

Mr. Sterin T Jose's focus at Kalkitech is on leading customer project/product proposals based upon requirements, including estimating the timetable, design of the solution/system and resource scheduling. He also oversees the project development including technical and product management as well as project management and coordination with the customer. He has more than 13 years of experience in the design and development of grid communication protocols in addition to project management expertise. He has been the instructor for numerous technical courses related to grid communication protocols. Mr. Jose holds a Bachelor of Engineering in Electronics and Communication Engineering degree Calicut University – NSS College of Engineering, Kerala, India.