Alejandro Schnakofsky, ABB Inc.

Digitizing copper
Defining key elements to ensure a successful integration from concept to maintenance

Abstract—in this paper, IEC61850 GOOSE message is presented as a vehicle to replace inter bay copper connections. The author hopes for this to become a guide for projects that outlines elements that must be considered to ensure the successful integration of this new technology. Sample applications are also provided to describe this new approach to Protection and Control (P&C) design.

I. Nomenclature

- P&C – protection and control
- GOOSE – generic object oriented substation event
- GCB – goose control block
- VLAN – virtual local area network
- IEC – International Electrotechnical Commission
- IED – intelligent electronic device
- SCL – Substation Configuration Language
- SAS – substation automation system
- MAC address – media access control address
- GoCB – GOOSE Control Block

II. Introduction

Six years have elapsed since the introduction of IEC 61850 to the world. The standard addressing communication networks and systems in substations has been accepted and applied worldwide in a number of projects in both transmission and distribution systems. Not just a communication standard, IEC 61850 brought radical changes and new challenges to Protection and Control design. Like any other new idea, concept or practice, sharing the experience of those that have journeyed into the unknown is of great importance for them to be widely accepted and grow into "the way to go". IEC 61850 increased acceptance comes as a result of real successful operating installations thanks to the work and collaboration of consultants, end users, and manufacturers. Philosophies and best practices have been developed and adopted, but not widely shared leaving those starting a project with the need to look for advice or reinventing the wheel. Not doing so could potentially result in a project failing to capitalize on the benefits and engineering plasticity offered by IEC 61850. Generic Object Oriented Substation Events (GOOSE) is a specific area where engineering and operation challenges exist, but have been resolved in installations in different ways addressing arrays of particular needs. GOOSE messages have emerged as a valuable tool in P&C design allowing engineers to easily deploy breaker failure, automatic bus transfer, bus protection schemes, DFR, alarms, interlocking, blocking, and control schemes in a cost effective and easily editable manner. The flexibility provided by GOOSE allows designers to implement more complex schemes thanks to the absence of cost constraints since schemes are independent of costly copper connections. This paper intends to shed some light into the challenge of applying GOOSE messages in an effort to "digitize copper" in P&C design; it attempts to become a valuable reference for P&C engineers considering application of GOOSE messages based on lessons learned, experience, and the standard itself. The paper will be divided in the following components; first, background information regarding the standard will be provided. Second, a detail description of the GOOSE message will be provided including GoCB, performance, Ethernet datagram, and triggers. Third, recommendations regarding the engineering of the system are presented. Finally, suggestions regarding the validation of the implementation are given.

III. IEC61850 communication networks and systems in substations

IEC 61850 was introduced to the industry as a joint effort among product vendors, consultants, and end users. It seeks to consolidate the knowledge acquired by the industry to generate an open and standard protocol with device interoperability as its main goal. The standard defines a data model (and a language to describe it) to illustrate all the functions and equipment that are available inside a substation. It also defines a series of services that interface with the data model to perform P&C and SAS related functions. It then defines the mapping of the data model and services to existing communication protocols (MMS, ISO/IEC 8802-3). The first edition of the different components of the standard was published in 2003-2004. The standard is structured in different parts (Fig. 1). The idea is that changes in one part may or may not affect other parts of the standard in an effort to make the initiative friendlier to changes in technology. From a data modeling and communications perspective, the standard defines the ingredients needed for interoperability.

A. SCL and the data model

IEC 61850 introduces a self described object oriented naming convention that defines each element and function that is performed inside a substation. This data model standardizes the implementation of the protocol, eliminating the need to cross reference memory locations, registers, or indexes to their actual meaning in the substation domain. A language for the purpose of documenting this data model was developed; as well as standard files which enable the exchange of information among the tools needed to configure the system as well as the programming of end devices e.g. IEDs.

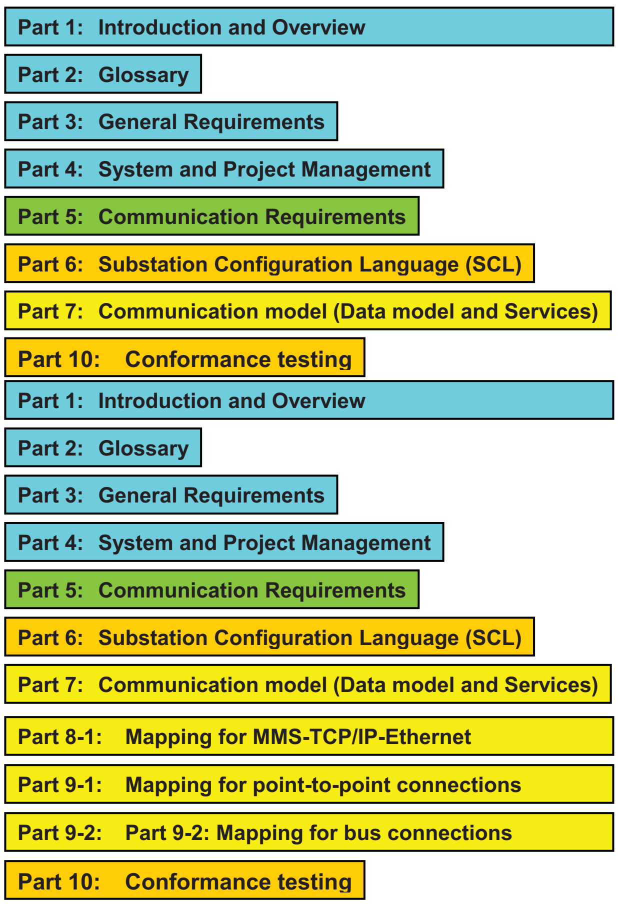

Fig. 1. Parts of 61850

The data model offers an intuitive description of functions (Autoreclosing, overcurrent protection, etc) and primary equipment (transformer, circuit breaker, etc) inside a substation. Each function and equipment is defined as a logical node which is an object that is defined by the data it holds and its methods. Some examples include:

- PDIS: Distance protection
- CSWI: Switch controller
- PTOC: Time overcurrent protection
- XCBR: Circuit breaker
- RREC: Autoreclosing

Each logical node is made up of data objects, and data objects are made of data attributes (the possible values of a data objects and what actually represents a value or state). Here is the example of XCBR:

- XCBR (logical node for circuit breaker)
- Loc (data for local operation)
- OpCnt (data for operation counter)
- Pos (data for position of the breaker)
- BlkOpn (data for indication of blocked open operation)
- BlkCls (data for indication of blocked close operation)

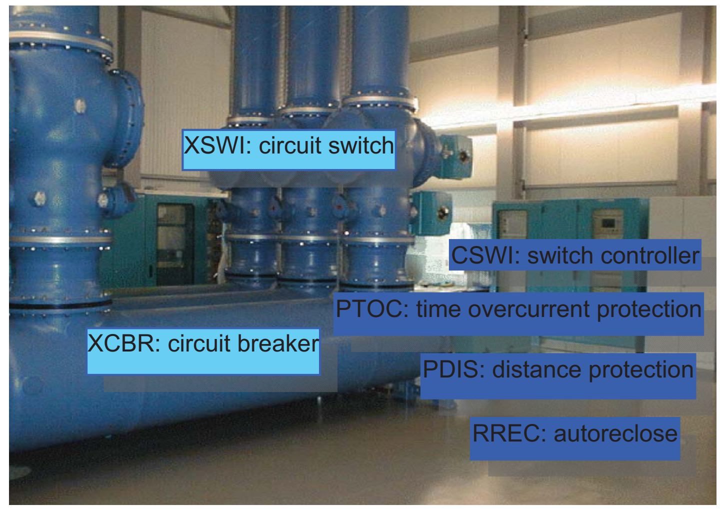

Fig. 2. Logical nodes in a substation

Each one of the data objects that make up XCBR have values associated with it. These values are called data attributes and represent the actual information of the function such as an analog measured value, double status point, single status point, etc. Going back into the XCBR example the following would be data attributes of Loc and Pos:

Loc
- stVal (status TRUE or FALSE)
- q (quality of the data attribute)
- t (time stamp of when the information was updated)

Pos
- stVal (status ON, OFF, INT, or BAD)
- q (quality of the data attribute)
- t (time stamp of when the information was updated)

Looking at the logical node we now have the status of whether the circuit breaker is in local or remote (XCBR.Loc.stVal), and the time stamp of the information (XCBR.Loc.t). The position of the circuit breaker can be obtained in a similar way (XCBR.Pos.stVal). This self describing model is a significant step forward from other protocols were the definition of data was based on memory locations and the definition of such data was left to be defined by the different manufacturers which resulted in a large number of different implementations making integration a difficult task.

00001: <0>	00014: <0>	00027: <0>	00040: <0>	00053: <0>	00066:
00002: <0>	00015: <0>	00028: <0>	00041: <0>	00054: <0>	00067:
00003: <0>	00016: <0>	00029: <0>	00042: <0>	00055: <0>	00068:
00004: <0>	00017: <0>	00030: <0>	00043: <0>	00056: <0>	00069:
00005: <0>	00018: <0>	00031: <0>	00044: <0>	00057: <0>	00070:
00006: <0>	00019: <0>	00032: <0>	00045: <0>	00058: <0>	00071:
00007: <0>	00020: <0>	00033: <0>	00046: <0>	00059: <0>	00072:
00008: <0>	00021: <0>	00034: <0>	00047: <0>	00060: <0>	00073:
00009: <0>	00022: <0>	00035: <0>	00048: <0>	00061: <0>	00074:
00010: <0>	00023: <0>	00036: <0>	00049: <0>	00062: <0>	00075:
00011: <0>	00024: <0>	00037: <0>	00050: <0>	00063: <0>	00076:
00012: <0>	00025: <0>	00038: <0>	00051: <0>	00064: <0>	00077:

Fig. 3. Modbus sample network traffic (read coils (1-78)

It is important to emphasize that GOOSE messages allow the sharing of groups of data attributes (a breaker is open or closed; a protection element is picked up or not, etc). These groups of data attribute which are utilized by GOOSE messages are called datasets. The Substation Configuration Language is used to describe the logical nodes (functions) that are available inside a particular IED. It also allows the documentation of the allocation of such functions in the one line diagram of the substation. It additionally contains information regarding the communications network structure, time servers, protocols gateways, etc. The language (as of edition 1 of the standard) has four files that enable the sharing and documentation of information regarding IEDs and the SAS.

- SSD: Substation Specification Description
- SCD: Substation Configured Description
- CID: Configured IED Description
- ICD: IED Capabilities Destination

B. Communication services

The standard defines several communication services. These services are utilized to share, control, and modify data. The following are the most relevant:

- Sampled Measured Values: service used to publish synchronized raw analog values
- GOOSE: generic object oriented substation event utilized to publish status points, counters and unsynchronized analog measured values
- Time Synch: service utilized to provide IEDs with a common time reference
- Client/Sever services: these include control model, substitution model, settings group model, reporting and logging model, file transfer, and discovery (self description)

IV. GOOSE message in detail

GOOSE messages provide the possibility for high speed and dependable distribution of information over a communications network. It represents the transmission of data attributes (position of a breaker, pickup status of a protection function, etc.) from a source (publisher) to a destination (subscriber). The publisher sends this information to the network and the subscriber(s) in turn receives the information. The information received (data attributes) is then used by the subscriber as a variable to perform a function. One example is a breaker failure scheme where the publisher is an IED that publishes its trip signal (e.g. PTOC.Op.general). The breaker failure relay is subscribed to this data attribute and internally utilizes this information to initiate a pickup timer that if consumed will in turn operate the breaker fail scheme. This chain of events is not different than the chain that occurs in a conventional scheme where the exchange of information is accomplished via physical contacts and copper connections.

A. Communication characteristics

This mechanism operates as an Ethernet (802.3) multicast in an effort to achieve the highest performance. Typically, an Ethernet multicast is a message that is forwarded to all the ports in a switch with the exception of the port that generated the message (and those that are not part of the VLAN if applicable). GOOSE messages are imbedded inside an Ethernet frame, all the information regarding the state of the data attributes is found inside the APDU (Application Protocol Data Unit). Since the frame is a multicast message, IEDs filter the information based on the destination multicast address and the Application Identifier (APPID). Its important to understand that GOOSE messages are not session dependent meaning that there is not presentation or association from a communications perspective needed to start receiving or sending the subscribed/published data.

Figure 4 shows the definition of each one the elements of an Ethernet frame. It must be noted that the maximum payload (size) of the APDU (where the data attributes reside) is 1492 Bytes. This number should be taken as a reference to the amount of information that can we shared via a single GOOSE message.

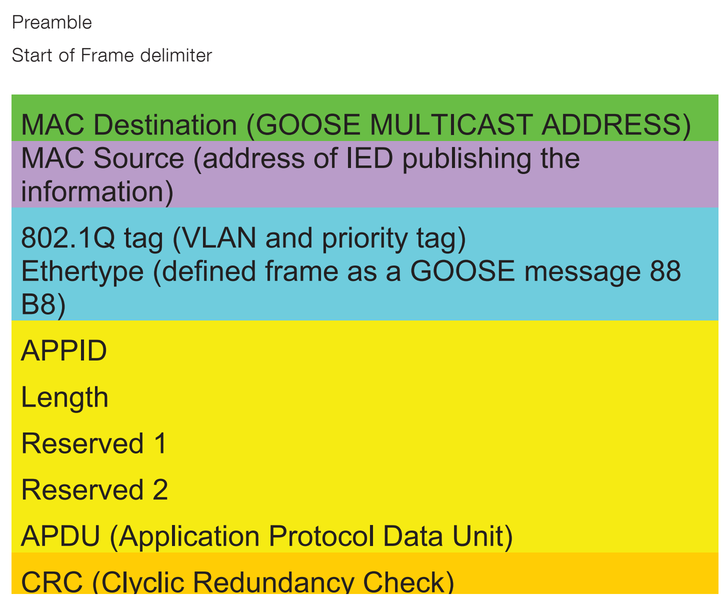

Fig. 4. 802.3 Ethernet frame

From an addressing perspective it can be seen in the frame that both a destination and a source address are included. The source address is the MAC address of the IED that is publishing the GOOSE message. The destination address is a multicast address that must have the following characteristics:

– First 3 octets must be: 01-0C-CD
– Second 2 octets must be from 01-00-00 to 01-01-FF

A sample destination MAC address would be 01-0C-CD-01-00-0D.

The frame also includes 4 bytes for priority tagging and VLAN id, 2 bytes for Ethertype (88B8), 2 bytes for APPID, 2 bytes for length, and up to 1492 bytes of data.

As previously mentioned, APPID is used by the subscribers to disregard messages that do not apply to them. The APPID has a value range from 0000 to 3FFF. The 0000 value represents lack of configuration. It is required to use a unique APPID per publisher to easily match frames to their publisher. From a performance perspective it must be noted that IEC 61850-5 defines two maximum transmission times for fast GOOSE messages. The standard defines 3 classes; P1, P2, and P3, where P3 has the highest performance requirements. Up to half a cycle is allowed for class 1 type 1 messages. Up to a quarter of a cycle is allowed for class 2/3 type 1 messages. Type 1 messages are simple messages that represent a trip, close, hold reclose, etc signal. The transmission time is the total time it takes to transmit, travel through the network, and receive the message making it analogous to the time that it takes for a device to pickup an output contact and for that output contact to be sensed by the physical input of the associated device.

B. The GOOSE Control Block (GoCB)

This control block holds the settings and manages the update of data of a GOOSE message in an IED that is publishing. There is one GoCB per GOOSE message and each has its own settings and address definition as follows:

GoCBName: name of the GoCB which must be unique

AppID: application identification which defaults to the path of to the block e.g. LD0.LLN0.GoCBName. In should be considered to set this to a unique value per publisher to easily identify the publisher inside the network traffic

Dataset: name reference to a created group of data attributes. Changes to the status of any of the data attributes that are part of the data set will result in the delivery of a GOOSE message

ConfRev: configuration revision; which changes any time there are any editions to the selected data set

NdsCom: needs commissioning, it’s used to indicate the GoCB needs further configuration

DstAddress: the destination address of the GOOSE message; it include the multicast MAC address, APPID, priority tag, and VLANID

MinTime: minimum amount of time that needs to expire prior to sending another GOOSE message

MaxTime: maximum amount of time that subscribed devices will wait to receive a message from the publisher before determining that the link is lost

The GoCB has the task of sending the GOOSE message per the above settings once a change of state in any of the variables that form the selected dataset is detected. Since this is a unidirectional message without confirmation, reliability is built in to the exchange by repeating the message several times. The messages are repeated by gradually increasing the time internal between them. A sample sequence would be as follows: upon the change of state of any of the data attributes inside the dataset the IED will send 3 messages spaced out by the time specified in the MinTime setting, the forth message will be spaced out by twice the MinTime setting, and the fifth one will be sent prior to the MaxTime setting.

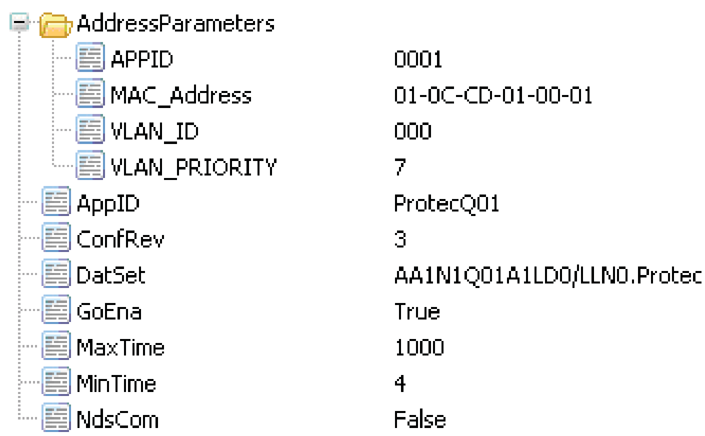

Fig. 5. GoCB settings

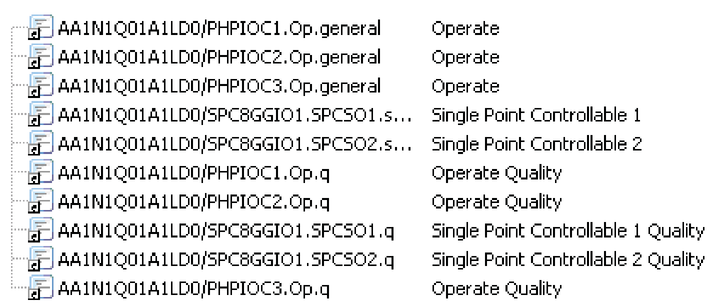

Fig. 6. Dataset Protec

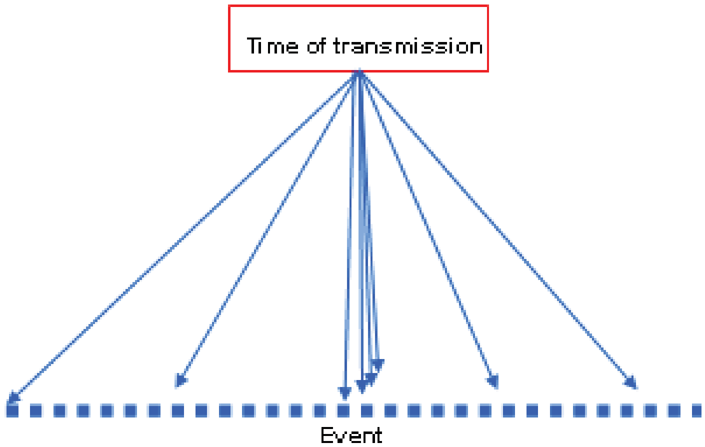

Fig. 7. Example of Goose message transmission

In summary, a GOOSE message is configured by creating a dataset and then completing the settings of the GoCB which require the selection of a multicast address, APPID, AppID, MaxTime, and Min-Time. Some tips regarding the setting of the GoCB include:

- Verify if the devices to be used require a particular syntax in the AppID, and APPID fields
- Use names for the GoCB and Datasets that are related to the application for which they are being used. This helps when performing the engineering and avoids confusion
- Verify the maximum amount of dataset entries you can have per GoCB
- Utilize priority tag to prioritize traffic (7 is the highest priority)
- Utilize VLANS to segment traffic among group of IEDs to make better use of the network's bandwidth

V. GOOSE ENGINEERING

The engineering of a P&C or SA system utilizing GOOSE messages presents new challenges to our industry. However, applying GOOSE messages to replace copper connections has the following benefits which must be evaluated:

- The process is self documented as it yields a file that describes the functions that each IED is performing, their location in the substation relative to the one line diagram, and the information that is being published and subscribed by/to IEDs
- The system can be easily changed since it only requires modifications that do not involve any physical hardware changes given that the copper connections are now communication messages
- The connections via GOOSE message are self monitoring, a break in a connection will result in the subscribed IEDs (the ones that are using the information) to initiate an alarm
- Replacing copper connections with GOOSE messages reduces the panel/enclosure space requirements otherwise needed to accommodate the bundles of wire
- GOOSE messages can be recorded and monitored with network analyzers, this serves as an additional piece of information that can be utilized to trouble shooting or post fault analysis purposes

There are several areas where GOOSE messages can be applied. Any scheme that requires the sharing of information between IEDs is an opportunity to use this technology. As previously mentioned, a breaker fail scheme is one that can be further explored.

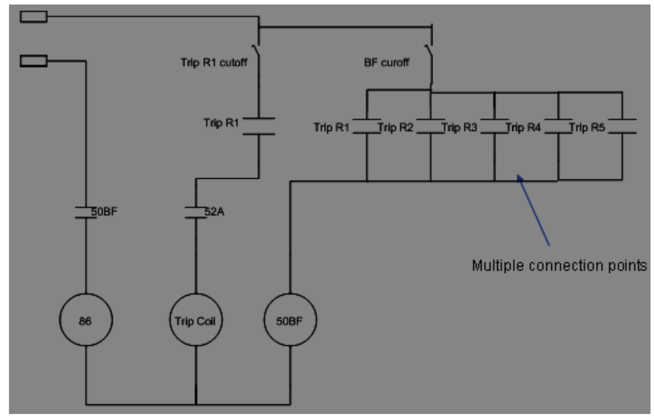

Fig. 8. Conventional breaker fail scheme

As seen in Figure 8, a conventional breaker fail scheme is made up of 2 basic elements; an initiation signal, which comes from the trip signal of the IED(s) that are part of the scheme, and a pickup timer which resides inside the breaker fail relay. If this timer expires, the breaker fail relay will perform its pre programmed sequence. The initiation of this timer is possible due to the copper connection that exists between the trip (or breaker fail initiate) output and the physical input of the breaker fail relay. This association can be described as:

Brk Fail Initiate = Physical Input 1

This copper connection can be replaced by a goose message. In general, the engineering process to create a connection via a GOOSE message has the following steps:

- Define the scheme: breaker fail
- Define the output variables that need to be published: in this case we will use PHPIOC1.Op.general which is the trip signal of the 50P-1 element
- Create and configure the GoCB in the IED that is publishing the signal
- Subscribe the breaker fail IED to the information that is being published
- Map the signal from the published GOOSE message to the Brk Fail IN in the breaker fail IED
- Send the configuration to both IEDs

In general this process isn't much different conceptually than the one required to make a copper connection. Looking at the steps it's important to define where each is taking place (refer to Fig. 9) Steps 1 and 2 and occur in the conceptual stage of the scheme. Step 2 is of great importance since the selection of signals is dependent on what is available inside the IED (IED data model).

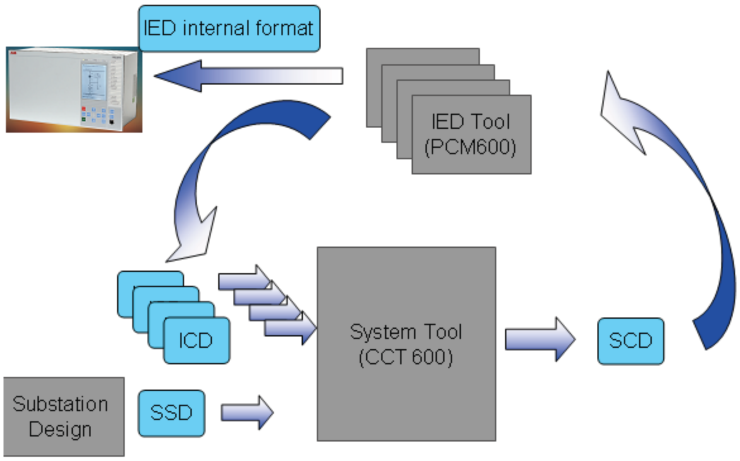

Fig. 9. Summary of configuration process

Step 3 occurs in the System Tool where the data set and GoCB are created (see Figure 5 and 6). Step 4 also occurs inside the System tool and here is when the GOOSE message is defined as an “input” to the breaker fail IED (see Figure 10). Once these steps are completed, the System tool yields a Substation Configured Description file. This is a file based on SCL that describes the configuration of the substation. In this example it is quite simple and it would only include the data model of the publisher and subscriber, the created GoCB (publisher) and inputs (subscriber), and the location of these 2 IEDs in the one line diagram of the substation.

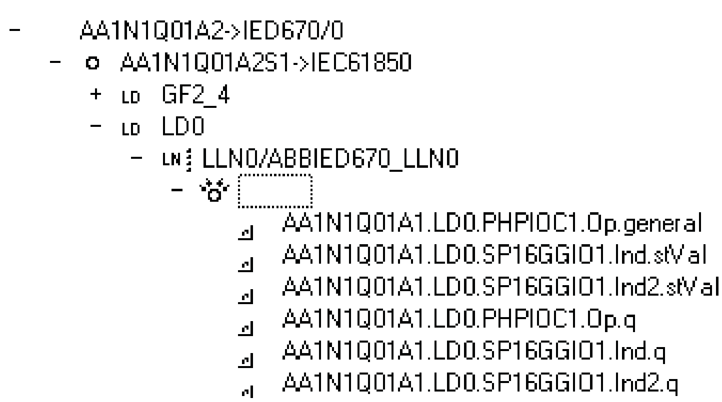

Fig. 10. Inputs to IED

Step 5 takes place inside the IED tool from the different manufacturers that are available. Instead of mapping a physical input to an element inside the IED, the information that the IED is subscribed to is available to map to internal variables inside the IED (see Figure 11).

	- Ied: AA1N1Q01A2, Logical Device: LD0	Ied: AA1N1Q01A2, Logical Device: LD0			
		SP16GGIO1		PHPIOC1	PHPIOC2
		DO: Ind, DA: stVal	DO: Ind2, DA: stVal	DO: Op, DA: general	DO: Op, DA: general
Brk Fail IN				x	
52A Brk A4		x			
52B Brk A5			x		

Fig. 11. Mapping of signals at the breaker fail IED

Step 6 is the final step of the process and it is when the configuration is sent to the IEDs.

There are several other applications that can be developed from this process, as stated, all this represents is a way to share information; the information that is shared (data attributes inside the dataset), and the purpose of it are questions that need to be answered based on the scheme that needs to be implemented.
The author would like to highlight three factors that are very important to consider for the engineering of GOOSE messages:

A. The System Tool Approach

The configuration of datasets, GoCBs, and inputs should be performed under the same tool in such a way that the output of the tool is an SCD file. This approach ensures that a single and accurate piece of information describing the system is available. This file should be treated as part of the documentation of the substation since GOOSE messages represent a connection just like a copper wire does. As such, here is where a test engineer will look to determine which device is publishing and which devices are subscribed to this information. Compatibility between the selected system tool and the IED tools must be ensured to avoid replication of work. The author has experienced instances where IED tools are not capable of reading an SCD file. In these cases, the CID (configured IED description) files from each IED must be imported individually into the environment. Next, it is necessary to redo the definition of the inputs into the specific device. This constitutes a duplication of effort where additional steps need to be taken.

B. IED data model and GOOSE messages

As shown in Figure 11, the GOOSE inputs are made available by the IED tool to associate them to internal variables. In this example it is clear that the GOOSE input is PHPIOC.Op.general which is the trip signal of the 50P-1 (PHPIOC LN) element as previously mentioned. It is intuitive and self describing which facilitates the mapping of that GOOSE input to any internal variable. In contrast, if that same status point was represented by a generic LN the process will require an additional reference sheet. This reference sheet will document what the data attribute of the generic LN represents. This reference will be needed for the mapping of GOOSE inputs to internal variables as well as for troubleshooting purposes. The author strongly believes the data should be published using the actual functions (LN) that generates it in the P&C domain rather than using generic LNs (GGIO), e.g. using PHPIOC.Op.general as opposed to SP16GGIO1.Ind.StVal

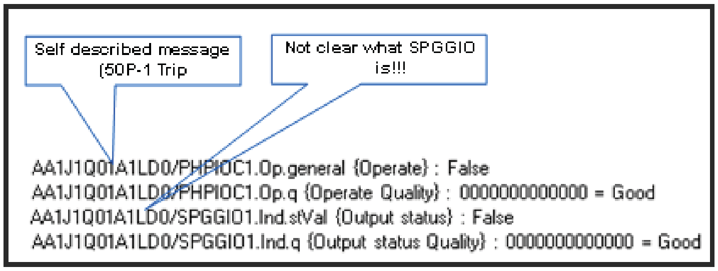

Fig. 12. Use LN that generates the signal in the P&C domain

VI. Testing and maintenance

The application of GOOSE messages to replace copper wires (digitizing copper) requires the preparation of testing and commissioning engineers to enable them to perform their work. This preparation should include both training (theoretical and practical) and well as equipping them with the necessary tools to test, validate and troubleshoot GOOSE messages.

Since this may be a new technology being applied for the first time it is important that all parties involved have a basic understanding of how it works. Field personnel are of most importance due to the fact that they carry the pressure of validating the scheme and often require making changes. The author recommends that training programs must include:

- Primer of Ethernet network theory
- Overview of IEC 61850 including a complete description of the data model of the IEDs being applied, dataset engineering, GoCB engineering, GOOSE input engineering, and GOOSE message performance expectations
- Detailed training on system, IED, and troubleshooting tools

The second component of the preparation is to have a troubleshooting tool. Conventional copper connections can be verified with a multimeter. The question often arises, how are goose messages verified? This is an important factor when evaluating vendors as most IEDs will have built in diagnostic screens that verify the connection. In addition to this, a replacement to the multimeter must be obtained; for the validation of GOOSE messages a network analyzer (commonly known as a "sniffer") must be an integral part of the toolbox. Network analyzers make the visualization of the of GOOSE messages possible. This information is key to determine the state of the data attributed being transmitted, configuration errors, and if the messages are being published at all.

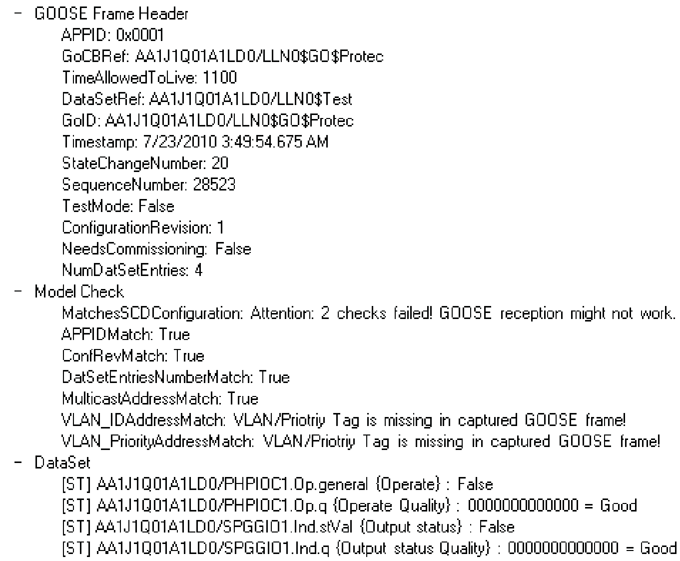

Fig. 13. Network analyzer used to validate GoCB configuration and status of data attributes

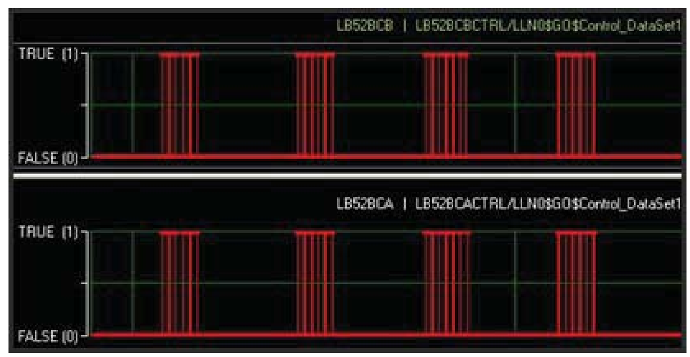

Fig. 14. Network analyzer used to visualize the status of data attributes

VII. Conclusion

GOOSE messages are a viable alternative to copper connections between IEDs. The power of the IEC 61850 model should be utilized to it potential, and thanks to it, signals that are self described are possible to be transmitted thus eliminating the need for a reference sheet. Almost any P&C or SAS scheme that requires the exchange of an output contact can use GOOSE messages to digitize the copper. Careful attention must be given to the selection of IEDs as not all implementations are equal. Finally, the need for training and testing tools cannot be underestimated; they are the base of a successful integration.

REFERENCES

1) "Communication networks and systems in substations", IEC standard 61850.

2) "Substation Automation Handbook", K. Brand, V. Lohmann, W. Wimmer.

Contact us

ABB Inc.
Distribution Automation
4300 Coral Ridge Drive
Coral Springs, Florida 33065
Phone: +1 954 752 6700
Fax: +1 954 345 5329

www.abb.com/substationautomation

1VAC4501-WP Rev A June 2012