Please cite as: J. Xu, C.-W.Yang, V. Vyatkin, S. Berber, Towards Implementation of IEC61850 GOOSE Messaging in IEC61499 Environment, IEEE Conference on Industrial Informatics (INDIN'13), Bochum, July 29-31, 2013

Towards Implementation of IEC61850 GOOSE Messaging in IEC61499 Environment

Jiang Xu[1], Chen-Wei Yang[1], Valeriy Vyatkin[2], Stevan Berber[1]
[1]Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand
[2]Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden
jxu877@aucklanduni.ac.nz, cyan083@aucklanduni.ac.nz, valeriy.vyatkin@ltu.se, s.berber@auckland.ac.nz

Abstract— **IEC 61850 is the new standard for substation automation. One of the IEC 61850 communication services, designed specifically for protection within the substation automation system is GOOSE messaging. GOOSE messaging is a peer to peer communication protocol aiming to satisfy the stringent communication requirements within protection systems operating over the Ethernet communication framework. One of the on-going research topics in this area is extending capabilities of IEC61850 to represent control logic by combining it with component software architecture of the IEC61499 standard. The Intelligent Logical Nodes (iLN) architecture is one such solution which combines the two standards. This development led to the need to implement communication services of IEC 61850 by means of IEC 61499. This paper presents such an implementation of IEC61850 GOOSE messaging using IEC61499 Service Interface Function Blocks implemented by means of the WinPCap library.**

Keywords-component; IEC 61850, IEC 61499, Function Blocks, Substation Automation System, intelligent logical node, Dataset, GOOSE Messaging, WinPCap Introduction

I. INTRODUCTION

The current infrastructure for the generation and distribution of electricity is dominated by centralized power grid architecture. The next generation of the so called smart grid will move away from the rigid centralized power grid architecture to a more distributed model as Distributed Renewable Energy Resources (Photovoltaic, Wind and Solar etc) and Distributed Energy Storage Devices (Battery, hydrogen storage and Plug-hybrid electric vehicles) are incorporated into the electricity grid [1].

One of the cornerstones of the future smart grid is the integration of the communication infrastructure into the electricity grid. The characteristic of the digital communication profile for the smart grid must be scalable, highly reliable, secure and high performance. In a distributed system, the communication framework has a huge responsibility in maintaining the integrity of the distributed control systems as the distributed based control algorithms rely heavily on the communication framework for gathering and exchanging of control signals between physical devices. The injection of bi-directional communications and two-way flow of electricity into the power grid allows both the consumers and the utilities to make smarter energy choices by becoming a part of the electricity generation and distribution process empowered by the delivery of real time data. The next generation of power grids would function like an energy internet [1] which would allow plug-and-play like functions for the distributed renewable energy resources and distributed energy storage devices.

IEC61850 is a new standard in substation automation design which aims to bring standardization to communication and requirement capturing in substation systems. The standard was initially introduced for communication within substation automation systems only, but has since extended to other domains including communication for wind turbines and control of hydro power plants [2]. IEC61850 brings Object Oriented approach to substation automation design by decomposing substation automation equipment or functions into functional objects known as logical nodes. IEC61850 introduces an array of modern communication profiles which includes GOOSE messaging for peer-to-peer communication and Server/Client services over the TCP/IP protocol operating over the Ethernet. Benefits of IEC61850 based systems include interoperability between IED devices, reusable system designs and system level design to substation automation design. What IEC61850 does not standardize however, is the logical implementation of the protection schemes. One standard which is capable of filling this missing void is the open standard of IEC61499.

IEC61499 is an open standard for distributed control and automation. IEC61499 introduces system level design for control, which is in line with the distributed design paradigm of IEC61850 Substation Automation systems. IEC61499 is of interest as it shares several similarities to IEC61850 which includes system level design of distributed systems (IEC61850 for communication and IEC61499 for control) and the encapsulation of functionalities (IEC61850 as logical nodes and IEC61499 as basic/composite function blocks).

This work is an extension of the work started in [3], which aims to replace the UDP based GOOSE messaging function blocks with data link layer communication sockets to achieve the functionality of IEC61850 GOOSE Messaging service. The focus of this paper is to show the implementation of IEC61850 GOOSE messaging using Service Interface Function Blocks (SIFB) developed for the ISaGRAF function block workbench and the interoperability of the SIFB solution with existing IEC61850 solutions. The function block solution is first demonstrated in a load shift case studies. Then interoperability tests are taken against an IEC61850 communication stack developed by SystemCORP.

The paper is structured as follows. Section II outlines the enabling technologies of the solution and expands on the use of GOOSE messaging in IEC61850. Section III expands on the implementation of GOOSE messaging as SIFB for the ISaGRAF workbench. Section IV expands on the results of the use case studies and section V presents conclusions and future work.

II. Contribution of the Paper

A. Enabling Technologies

Digital communication is one of the cornerstones of the envisioned future Smart Grid. IEC61850 [4] is one such standard which is aims to standardize and introduce modern communication technology to the substation automation domain. IEC61850 has already proven to reduce cost, improve efficiency and increase in reliability in substation automation systems. The standardization of communication protocols in the substation automation domain goes a long way in alleviating the frustration of incompatibility of devices when designing substation systems as manufacturers apply their own proprietary communication solutions for their devices due to the lack of a common communication solution. This has proven to be a huge stumbling block for the utilities when it comes to upgrading existing equipment to modern technologies as the entire equipment need to be replaced incurring huge costs in upgrading from existing equipment. GOOSE messaging is one of the many communication services in the IEC61850 standard that is of interest. GOOSE messaging was designed specifically to meet the stringent timing requirements that are required for protection systems.

Although IEC61850 standardizes the communication protocol of substation automation systems, it does not have guidelines on the implementation of logic for IEC61850 based systems. The TC57WG10 group, the technical group behind the standard, has shown interest in providing some sort of standardization or guidelines in the implementation of logic as part of the IEC61850 solution as shown in [5]. IEC61499 has been proposed as a viable solution to fill the void and there have been on-going research in this domain [6] which outlines the novel idea and the benefits of an IEC61850 and IEC61499 solution and several case studies [3, 5, 7, 8] have shown the feasibility of the combined solution. The case studies utilize the iLN architecture which demonstrates the capability of the combined IEC61499 and IEC61850 solutions. The iLN architecture utilizes the encapsulation property of function blocks artefacts to capture the information models of IEC61850 from the logical device level down to the common data types.

The work done in this paper highlights how well the two standards can be used from a data modelling and logic design point of view. The next step in the development of the iLN architecture is incorporating the IEC61850 communication services to the iLN design architecture.

B. IEC61850 GOOSE Messaging

GOOSE messaging is a peer-to-peer message passing service aimed to satisfy the stringent timing requirements necessary in protection applications. GOOSE messages is implemented over the link layer to reduce the overhead of the higher protocol layers introduced in the OSI stack to ensure that the protection schemes with strict timing requirements can be achieved. Examples of modern protection schemes designed using GOOSE messaging includes [9], where GOOSE messages play an integral part in exchanging critical information and data between several IEDs in the field to ensure the distributed protection scheme can function under the allowable timing constraints.

Currently, the first commercial IEC61499 function block vendor ISaGRAF, is actively working on a solution which combines IEC61499 with an IEC61850 protocol stack. The function block solution is built on top of the IEC61850 protocol stack which uses the function block component to handle the logic design and the IEC61850 component to handle the substation communication. The current solution has lots of potential, however, there is a slight disconnect between the design processes of the solution as the two components need to be configured separately. It would be better to have a solution which reduces the amount of time spent configuring the two components separately. A more seamless solution, which would allow the parameterization of the GOOSE messages to be done as part of the function block design process, would be more beneficial for the designers.

A viable solution is to develop SIFB for the ISaGRAF workbench to handle the GOOSE messaging service. This would allow the GOOSE messages to be configured as part of the function blocks design process.

III. IEC61850 GOOSE MESSAGING

A. Realiability of GOOSE Messaging

Protection applications in substation automation systems demand for a messaging service which is of dependable reliability and fast transmission speed. GOOSE messaging fills this need even though the messaging service is operating over the Ethernet framework which is considered to be nondeterministic. IEC61850 specifies several provisions to ensure the messaging service is dependable for protection applications. The first is the constant retransmission of GOOSE messages and the second is the management of received GOOSE messages by the subscribing devices.

GOOSE messaging is an unidirectional peer-to-peer multicast messaging service which does not receive any acknowledgement of whether the message has been received by the recipient nor is it desirable for the publisher to receive acknowledgement given the frequency at which the message is published and the number of subscribers to the message. The retransmission mechanism ensures the messaging service is reliable, even though no acknowledgement packet is ever received by the subscriber. It is assumed that the constant retransmission of GOOSE messages ensures at least one message will be received by its intended recipient in the required timeframe. There are two instances which trigger the sending of GOOSE messages. The first trigger is defined in the GOOSE parameter max time (*mt*). *mt* determines the interval of retransmission between consecutive GOOSE messages. The second trigger occurs when one or more of the dataset members change its value. In this case, the retransmission time is dependent on the time of transmission (*tot*) parameter. The initial time interval of *tot* will be short and will slowly increase to *mt* till the dataset is again updated with refreshed data.

As mentioned previously, GOOSE message is a unidirectional messaging service and does not receive any acknowledgement packets. Thus in situations where a loss of messages or loss of connectivity channels does take place, it is the responsibility of the subscriber to be resilient and handle these situations appropriately. Embedded within each GOOSE message frame is the time to live (*ttl*) parameter. The *ttl* variable specifies how long a GOOSE message should be active before the next message is received. The *ttl* time is calculated from the next *tot* by setting *ttl* as multiples of the next *tot* rather than equating to take into account of network delays [10]. On the subscriber side, the parameter time to wait (*ttw*) is equated from the *ttl* parameter from each GOOSE message. When *ttw* is greater than *ttl,* it is assumed either a message is lost or the communication channel is broken. The subscribing IEDs can respond by either changing its internal logic to accommodate the loss in packet or the communication channel, or raise predetermined alarms. The retransmission mechanism in this case is used as a keep-alive mechanism to check for the health of the communication channel.

B. Development of GOOSE Messaging SIFB

The GOOSE messaging SIFB was developed using the SIFB development kit provided by ISaGRAF. SIFB are written in the C Programming language using MS Visual Studio. The targeted ISaGRAF workbench for the SIFB application is ISaGRAF 5.2. The resultant SIFB is compiled into a .dll library executed by the ISaGRAF runtime during simulation.

The two socket libraries considered for the development of the GOOSE messaging SIFB were the Winsock (Windows Sockets) library and the WinPCap library. Winsock is a popular C library often used for the development of TCP/IP and UDP/IP based socket applications. However, Winsock was quickly eliminated from consideration as GOOSE messaging requires sockets developed for the Link layer of the OSI stack, which is not supported in Winsock due to security concerns. WinPCap is another free library which does support the development of link layer based sockets and is often used for packet capturing applications. However, WinPCap can be also used by programs to send, receive and filter over the data link layer. One successful and mature network protocol analysis tool, Wireshark was developed using the WinPCap library. WinPCap was thus chosen as the library for the development of the GOOSE messaging SIFB.

C. GOOSE SIFB Function Block – Publisher and Subscriber

The publisher SIFB is shown on the left hand side of Figure 1. The *INIT* input event signal initializes the GOOSE message frame, list the supported local network adaptor on the physical device and opening of the selected network adaptor for GOOSE messaging. The *REQ* event signal is received when the values of the GOOSE dataset members is changed. The input variables to the publisher function block are the GOOSE parameterizing variables. The GOOSE messaging service is configured with the input parameterizing values when the *INIT* event is received. Essential parametrical variables includes Destination (DES) and Source (SRC) MAC address of the publisher and the subscriber device, goCBRef (GREF), datSet (DS), goID (GOID), *mt* (AMAX) and the initial *tot* (AMIN). The ADT input is the content of the dataset that is published via the GOOSE message. The members of the dataset have been encoded into a single string variable separated by the '|' separator. Each data member is also tagged by a type string in front of the data to indicate the variable type of the dataset member with the ';' separator. The type tagging is used to ensure the dataset members are correctly encoded during transmission. E.g. for the ADT input "bool;data1|string;data2|double;data3", data1 is of type Boolean, data2 of type String and data3 of type Double.

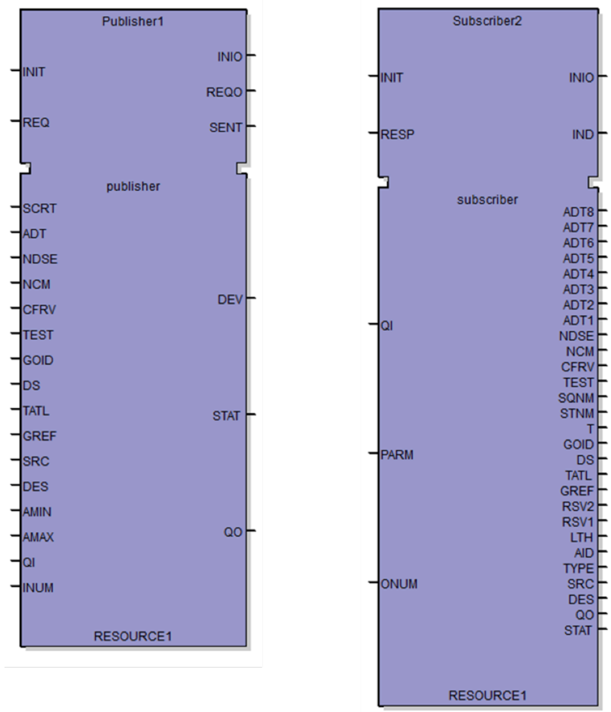

Figure 1. GOOSE Messaging SIFB - Publisher (Left) and Subscriber (Right)

The subscriber SIFB is shown on the right hand side of Figure 1. The *INIT* event signal operates similarly to its counterpart in the publisher SIFB. The *INIT* is the initialization signal and it is used to find, list and open the proper network adaptors. In addition, filtering is introduced on the subscriber to allow only GOOSE messages to pass through the subscriber. The *IND* output event is triggered every time a GOOSE message is received and the output parameters of the subscriber SIFB will be updated with the content of the GOOSE message. Essential output parameters include APPID (AID), timestamp of GOOSE message (T), the state number (STNUM), the sequence number (SQNUM), *ttl* (TATL), datSet (DS), GOOSE ID (GOID) and the App ID (APPID). The eight outputs ADT1-ADT8 are the SIFB outputs parameters for the members of the GOOSE dataset. The ADT output variables are of string type and is outputted each time the *IND* output event is triggered.

D. Challenges

One of the main challenges encountered during the development of the GOOSE SIFB is how the GOOSE messaging service could co-exist with the cyclic PLC scan cycle. SIFBs in ISaGRAF normally execute with the rest of the function block program during every PLC scan cycle. One scan cycle is defined as the time it takes for the PLC to:

1. Input Scan: Sequentially scan the input of the PLC device and update to the PLC memory often referred to as the input image table [11].
2. Program Scan: Execute programmed logic with the data in the input image table and update the output image table with the updated values [11].
3. Output Scan: Update the output modules with the values from the output image table [11].

The duration of scan cycles is usually in the magnitudes of milliseconds, but scan time does vary from PLC to PLC and is dependent on the device I/Os and size of the programmed logic. The potential pitfall of running the GOOSE message service as a part of the PLC execution cycle is illustrated in the top figure of **Error! Reference source not found.**.

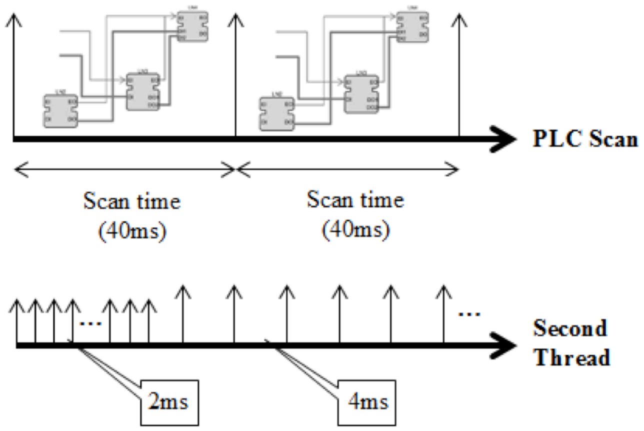

Figure 2. Multithreaded implementation for GOOSE Messaging in SIFB

If, for instance, the PLC scan cycle time is 40ms and the *tot* time of a GOOSE message is 4ms, then only one GOOSE message will be published per scan cycle. This is problematic as successive GOOSE messages will only be published approximately every 40ms, rather than the required 4ms. The introduction of multithreading to the SIFB would allow the transmission mechanism of GOOSE messaging to run independently from the PLC scan cycle as shown in Figure 3. There are two threads in the SIFB and a shared GOOSE message frame between the two threads. Thread 1 is tasked to obtain the updated content of the dataset values and Thread 2 is tasked with the transmission mechanic of the GOOSE messaging service. The GOOSE message frame is protected by a mutex to ensure mutual exclusion between the two threads as it is written or read by the two threads.

Error! Reference source not found. illustrates the overview of how the two threads co-exist in the SIFB. Thread 1 operates under the PLC scan cycle execution and obtains the updated GOOSE dataset values from the function block program as it is executed. When Thread 1 receives the *REQ* event and the associated updated GOOSE dataset value, Thread 1 will request access for the mutex to update the value to the GOOSE message frame.

Thread 2 is operating independently from the PLC scan cycle and thus is able to publish the GOOSE message at the required *tot*. Thread 2 is writing to the GOOSE message frame each time a message is published to update the GOOSE message parameters such as the SqNum (Sequence Number) and the StNum (State Number).

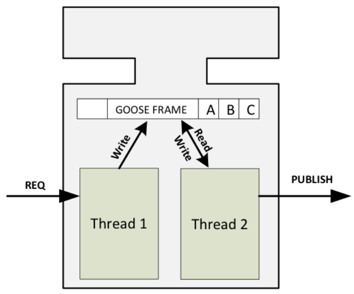

Figure 3. Multithreaded GOOSE Messaging SIFB

IV. Results

The GOOSE messaging SIFBs were tested under two use cases. The first use case demonstrates the GOOSE messaging SIFB used for a smart grid application simulated under the co-simulation environment. The second use case is an interoperability test between the GOOSE Messaging SIFB and an IEC61850 protocol stack on the DK61 Beck development kit from SystemCORP.

A. Load Shift Case Study

The smart grid application used is a load shift application shown in Figure 4. The aim of the load shift application is to balance the load between the two sources (1 and 2) as loads (grey shaded) are added to the system. The two main switches: Sectionalizer and the Tie switch are manipulated by the load shift controller as the loads are added to the system

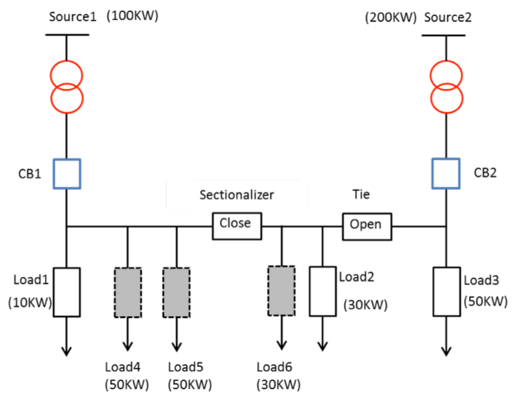

Figure 4. Schematic of Load Shift Application

The case study is simulated under the Co-Simulation environment [12] with the GOOSE messaging SIFB used as the messaging service to exchange the telemetry and the control data between the plant and the FB controller in the Co-Simulation environment. The load shift plant was developed in Matlab Simulink and the load shift control was developed in ISaGRAF. The Co-Simulation architecture with the GOOSE messaging SIFB is illustrated in Figure 5.

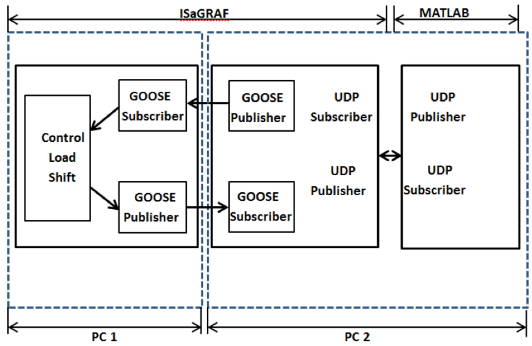

Figure 5. Communication architecture of Co-Simulation with GOOSE Messaging

The Co-Simulation environment is distributed between two PC. The MATLAB plant model sends the simulated telemetry data to a function block program on PC2 via UDP sockets. The function block program on PC2 uses the GOOSE messaging publisher SIFB to publish the telemetry data to PC1, where the load shift controller is located. The load shift controller on PC1 will process the telemetry data and send control signals back to the MATLAB plant if required. The control signal will be sent through a GOOSE messaging publisher on PC1, received by a GOOSE SIFB subscriber on PC2 before forwarding to the MATLAB plant model via UDP.

The result of the case study is shown in Figure 6. The graph shows the total power consumed by the loads on source 1 and source 2 as additional loads are added to the system. The initial load condition of the two sources is as follows. Source 1 has a total capacity of 100KW and is servicing a total of 40KW of load. Source 2 has a total capacity of 200KW servicing 50KW of load. The Sectionalizer Switch is in the close state and the Tie switch is in the open state. The graphs in Figure 6 show the resultant power readings on the two sources as these steps are taken:

1. Add load4 (50KW)

   When load4 is added, Source1 is now servicing a total of 90KW (10KW+50KW+30KW) of load. The switch positions of the Sectionalizer and the Tie switch remain unchanged as the system is still balanced as shown in Figure 6, number 2.

2. Add load6 (30KW)

   When load6 is added, the total consumption on source1 is now 120KW, which is over the allowable capacity on source1. The Sectionalizer switch will now open and the Tie switch will close. This will shift 60KW of load to source2. Source1 will now service 60KW (10KW+50KW) of load, and source2 will now service 110KW (30KW+30KW+50KW) of load as shown on number 3 of Figure 6.

3. Remove load6 (30KW), Add load5 (50KW).

   When load6 is removed and load5 is added, source1 is now servicing 110KW (50KW+50KW+10KW) of load and source2 is now servicing (80KW) of load. The load on source1 is now greater than the capacity of source1. The Sectionalizer switch will now change to the close state and circuit breaker CB1 will now change to the open state. This will shift all loads on the system to source2. Source1 now services 0KW of load and source2 is serving 190KW of load.

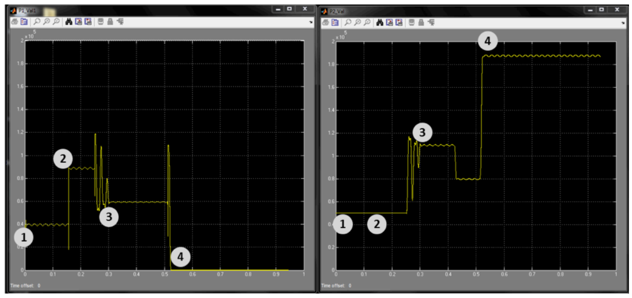

Figure 6. Results of Load Shift Application

The role of this use case demonstrate the success of the GOOSE messaging SIFB as the P2P communication medium for transferring telemetry data and control signals between the function block program and a plant as specified by the IEC61850 standard. The function block program on PC2 acts as the IEC61850 Analog Merging Units (AMU). The AMU is tasked with digitizing analog telemetries and publishing the sampled data to the IED devices for protection [13]. The AMU usually published telemetry samples to the IEDs via P2P Sampled Measured Values (SMV). However, since all the telemetries are aggregated from one source for this simulation environment, i.e. Matlab, GOOSE messaging is used to carry all the telemetry data in a single GOOSE message from the AMU to the controllers. The function block program on PC2 also acts as the receiving hub for control signals from the function block controllers. A single GOOSE messages contains control signals for all the circuit breakers in the Matlab simulated plant.

B. Interoperability test with IEC61850 Stack

The second case study is the interoperability test. The aim of this test is to check whether the GOOSE messaging SIFB is able to publish or subscribe to GOOSE messages from a third-party IEC61850 protocol stack.

The first test is to use GOOSE messaging to control four of the LED lights on the BECK device with the GOOSE SIFB publisher. The GOOSE SIFB will publish the values of the LED lights as GOOSE messages and the BECK device will receive the GOOSE message and update the four LEDs accordingly. The dataset of the GOOSE message contains four Boolean mapped to LED5-8 on the BECK device. LED5-8 will be on if the value of the corresponding Boolean is true, and off if the value is false.

Figure 7 shows the result of the test. The values of the four Booleans published will alternate between "false,true,false,false" and "true,true,false,false" every five seconds. On the BECK device, the combinations of the LED5-7 will alternate between "off,on,off,off" and "on,on,off,off" respectively.

Figure 7b shows the output of the LED when the GOOSE message "false,true,false,false" is received. LED6 is on and LED5,7-8 is off.

Figure 7c shows the output of the LED when the GOOSE message "true,true,false,false" is received. LED5-6 is on and LED7-8 is off.

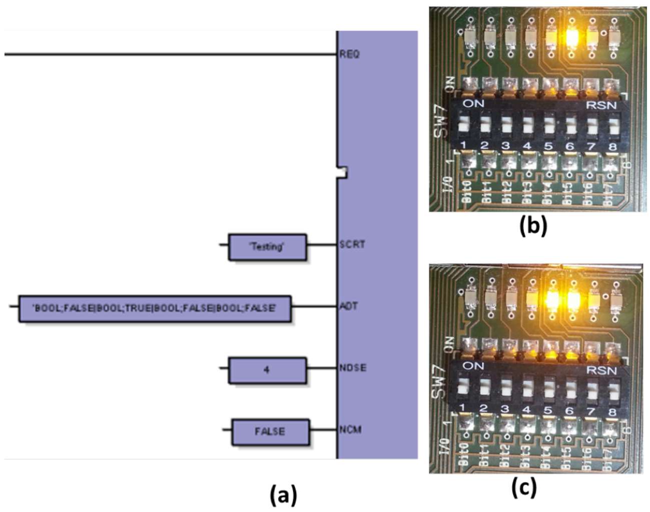

Figure 7. Interoperability Test: GOOSE SIFB as the Publisher , BECK as the subscriber

The second test is to have the BECK device act as the GOOSE publisher and the GOOSE SIFB as the subscriber. The BECK device will publish the status of DIP switches 5-8 in GOOSE messages. The status of the switches is represented by four Booleans. The on position will be represented by Boolean true and the off position will be represented by Boolean off. Figure 8 shows the results of the test. When the DIP switch 6 is toggled, published GOOSE message from the BECK device will be "false,true,false,false". The received message on the subscriber is shown in Figure 8b and c. The ADT outputs shown in Figure 8b is "0,1,0" representing the DIP switches 5-7. Figure 8c is the capture of the same GOOSE message from the wireshark program which also shows the "false,true,false" values for the DIP switches 5-7.

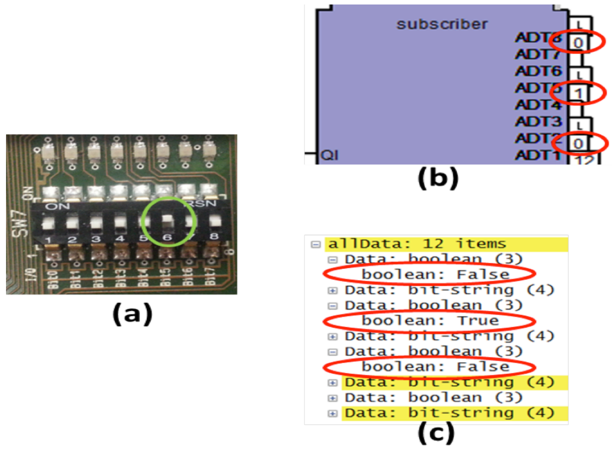

Figure 8. Interoperability Test: BECK as the Publisher, GOOES SIFB as the Subscriber.

V. Conclusion and Future Work

The paper presents the initial results of the IEC61850 GOOSE messaging service implemented as an IEC61499 SIFB. A GOOSE messaging SIFB provides a more seamless solution to the design of IEC61499 and IEC61850 applications. Results of this development have been validated in two case studies: one for a smart grid load shift protection application using the co-simulation environment and the other is interoperability test of the GOOSE SIFB with an IEC61850 protocol stack as both a publisher and a subscriber.

Future work towards a seamless solution for the combining IEC 61850 and IEC 61499 standards includes providing concrete timing results for compliancy to the IEC61850 standard and incorporating the configuration of SCL files into the function block design process which allows the import and export of GOOSE messaging configuration.

References

[1] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale, "The Future Renewable Electric Energy Delivery and Management (FREEDM) System: The Energy Internet," *Proceedings of the IEEE,* vol. 99, pp. 133-148, 2011.

[2] C. Brunner, "IEC 61850 for power system communication," in *Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*, 2008, pp. 1-6.

[3] Y. Chen-Wei, G. Zhabelova, V. Vyatkin, N. C. Nair, and A. Apostolov, "Smart Grid automation: Distributed protection application with IEC61850/IEC61499," in

Industrial Informatics (INDIN), 2012 10th IEEE International Conference on, 2012, pp. 1067-1072.
[4] International Electrotechnical Commission, "IEC 61850: Communication networks and systems for power utility automation," in *Part 7-4: Compatible logical node classes and data classes,* vol. IEC 61850, ed, 2010.
[5] Y. ChenWei, V. Vyatkin, N. K. C. Nair, and J. Chouinard, "Programmable logic for IEC 61850 logical nodes by means of IEC 61499," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 2011, pp. 2717-2723.
[6] Higgins N., Vyatkin V., Nair N., and S. K., "Distributed Power System Automation with IEC 61850, IEC 61499 and Holonic Control," presented at the Proc. IEEE Conference on Systems, Machine and Cybernetics 2008, Singapore, 2008.
[7] G. Zhabelova and V. Vyatkin, "Intelligent Logical Nodes of IEC 61850 and IEC 61499 for Multi-agent Smart Grid Automation," *IEEE Transactions on Industrial Electronics 2010,* 2010.
[8] V. Vyatkin, G. Zhabelova, Y. Chen-Wei, D. McComas, and J. Chouinard, "Intelligent IEC 61850/61499 logical nodes for smart metering," in *Energy Conversion Congress and Exposition (ECCE), 2012 IEEE*, 2012, pp. 1220-1227.
[9] A. Apostolov and B. Vandiver, "IEC 61850 GOOSE applications to distribution protection schemes," in *Protective Relay Engineers, 2011 64th Annual Conference for*, 2011, pp. 178-184.
[10] International Electrotechnical Commission, "IEC 61850: Communication networks and systems in substations," in *Part 8-1: Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3* vol. IEC 61850, ed, 2010.
[11] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*: Prentice Hall, 2007, 2007.
[12] V. Vyatkin, G. Zhabelova, N. Higgins, K. Schwarz, and N.-K. C. Nair, "Towards intelligent Smart Grid devices with IEC 61850 Interoperability and IEC 61499 open control architecture," in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, 2010, pp. 1-8.
[13] A. Apostolov. (December 2011). *Impact of IEC 61850 on Bus Protection*. Available: http://www.pacw.org/issue/december_2011_issue/communication/impact_of_iec_61850_on_bus_protection.html.