Please cite as: G. Zhabelova, S. Patil, C.-W. Yang, V. Vyatkin, "Implementation of Smart Grid Intelligence Applications using IEC 61499 Reference Architecture", IEEE Conference on Industrial Informatics (INDIN'13), Bochum, July 29-31, 2013

Smart Grid Applications with IEC 61499 Reference Architecture

Gulnara Zhabelova, Sandeep Patil, Chen-wei Yang
Electrical and Computer Engineering
The University of Auckland
Auckland, New Zealand
gzha046@aucklanduni.ac.nz, spat251@aucklanduni.ac.nz,
cyan083@aucklanduni.ac.nz

Valeriy Vyatkin
Department of Computer Science, Computer and Space Engineering
Lulea Tekniska Universitet
Sweden
valeriy.vyatkin@ltu.se

***Abstract*— This paper presents implementation of the Distributed Grid Intelligence (DGI) applications such as Intelligent Fault Management and Load Balancing using IEC61499 reference architecture. This enables system level design of distributed applications with a direct pathway to deployment to hardware. The use of IEC 61499 improves scalability, re-configurability and maintainability of automation software. Both applications were verified using co-simulation approach: control and power system simulated on PCs, and power system simulation on a PC networked with the number of distributed hardware running control algorithm. Both DGI applications were deployed to commercial programmable automation devices and embedded controllers. The use of IEC 61499 facilitates deployment of hardware independent function block model to the variety of compliant hardware. The paper also discusses distributed SCADA concept with IEC61499 implemented with the so-called Composite Automation Type (CAT) function blocks, which combine functional and visual components. These blocks are used to implement human-machine interface of the FREEDM Smart Grid demonstrator reacting on plugging and unplugging of energy resources.**

Keywords*— *FREEDM, IEC 61850, IEC 61499, intelligent fault management, load balancing, protection, energy management, GreenHub, distributed intelligence, plug and play, distributed SCADA.

I. INTRODUCTION

The Smart Grid concept is being widely researched and prototyped in the last decade. Investigation towards the sustainable and efficient energy delivery system, has identified two significant paradigm shifts [1]. The future infrastructure should be ready for wide spread of distributed renewable energy resources (RES) and bi-directional energy flow. In order to integrate components of the future grid such as DRER[1], PEV, AMI and DESD, the grid needs to be smarter. Smart Grid promises efficient, sustainable and reliable electricity infrastructure, where every member of the grid (consumer, energy retailer, operator and generation.) is an active participant in fast-paced real-time energy market.

The Future Renewable Electric Energy Delivery and Management (FREEDM) centre is part of the world research effort to realise Smart Grid. FREEDM centre envisions the backbone of future grid to be a new power distribution infrastructure, allowing for integration of Smart Grid components in a "plug and play" manner and enabling bi-directional flow of electricity. This new energy delivery infrastructure they call "Energy Internet" [1].

To control such a complex infrastructure, FREEDM system needs to perform functionality of energy and power management, energy dispatch, dynamic reconfiguration, power balancing, maintaining system stability and reliability, fault identification and location (self-healing), and managing business policies while maintaining security of grid participants. Within the FREEDM context, these functionalities constitute Distributed Grid Intelligence (DGI). To manage future Smart Grid, DGI has several applications such as Intelligent Energy Management (IEM), Load Balancing (LB) and Intelligent Fault Management (IFM).

Implementation of Energy Internet concept imposes several issues. FREEDM centre envisions the major challenge will be implementing DGI applications in a distributed manner across multiple execution platforms (i.e. IEM and IFM). Solution to the above problem, FREEDM expects to be the development of an open-source-standard-based software and communication platform to implement DGI [1].

In this paper we propose a cyber-physical approach to FREEDM centre's solution. We propose an open-standard based framework for implementing DGI applications, utilizing IEC 61499 industrial standard.

The job of managing wide-spread, highly occupied and task intensive systems is better suited for distributed control architectures. Distributed systems deliver reliability, concurrency, performance and scalability. IEC 61499 standard is designed for the development of distributed control and automation systems [2]. In power systems domain, IEC 61499 is capable of implementing automation, control and protection algorithms in a distributed manner, where multiple devices e.g. IEDs are able to communicate horizontally.

Future energy infrastructure with the distributed devices with plug and play capabilities will require a new concept of SCADA systems, where the devices come on- and off- line dynamically during its operation, and so the central control HMI needs to be updated. The SCADA system ought to handle dynamic re-configuration of the system. The paper presents discussion on utilising capabilities of IEC 61499 to realize the distributed SCADA concept. This implies that a distributed

[1]See the list of acronyms used in the end of the paper

device such as DRER or DESD will be enhanced with control algorithm - "intelligence", that is algorithm of its autonomous behaviour. Upon plugging/unplugging the device will communicate to the SCADA centre to update current state of the system on the HMI.

The structure of the paper is as follows. IFM and LB applications are introduced in section II. Section III presents development and implementation of IFM and LB systems using proposed framework. Section IV describes co-simulation of the developed applications and results. The concept of distributed SCADA is discussed in section V. Conclusion summarizes performed work and outlines future goals.

II. Distributed Grid Intellegence Applications

FREEDM system is a prototype of Smart Grid of reasonable complexity, with modular architecture. Devices such as DRER, DESD, Smart Meters and loads are grouped and managed by SST block - or DGI node. Several such SST blocks comprise one level of distribution system, governed by distributed intelligence - DGI. Another module within FREEDM system is an IFM device - new generation fault isolation device with intelligent control. Such highly distributed and modular architecture requires intelligent algorithms implemented in a distributed way [1]. The paper proposes and demonstrates the utilization of IEC 61499 for such applications as IFM and LB. In both cases corresponding FB network was developed mirroring the modular FREEDM system topology. Both IFM and DGI node were designed as separate modules - composite FBs, consisting of simpler modules - basic FBs. Such architecture is easy re-configurable, maintainable and scalable. System level development allows for top down design. Deployment of the system is straight forward.

The following sections will describe proposed implementation of DGI applications.

A. Intelligent Fault Management Application

FREEDM project has proposed a novel protection scheme [3]. The main concept is to divide the system into zones, using fault isolation devices (FID, new generation circuit breakers). Thus FID is situated at the borders of each section. The FREEDM protection strategy is shown on Fig. 1 using Green Hub simulation model. The protection scheme is divided into three zones and overall zone 0. Zone 0 protection is a backup protection for overall system. At each zone merging unit is placed at the terminal of distribution line and the feeder of the load to measure current, digitize and transfer the sensed values to IFM. Each zone has an IFM which runs the protection algorithms and incorporates into DGI – distributed grid intelligence. The overall protection scheme consists of primary and secondary protection.

Primary protection used is the differential scheme: if the sum of current in a zone equals zero, it indicates either there is no fault or fault is outside the zone of that IFM. In case the sum of the currents within a zone is not zero then the fault is within the zone and IFM makes decisions to trip the FIDs at the border of the faulty section. This protection algorithm mostly relies on IFM node, which in this case can be a computer or digital relay. The secondary protection for this system is overcurrent protection. In this case IFM examines each incoming current sample; then compares against pre-set current value, which can be 3-5 times of the rated current.

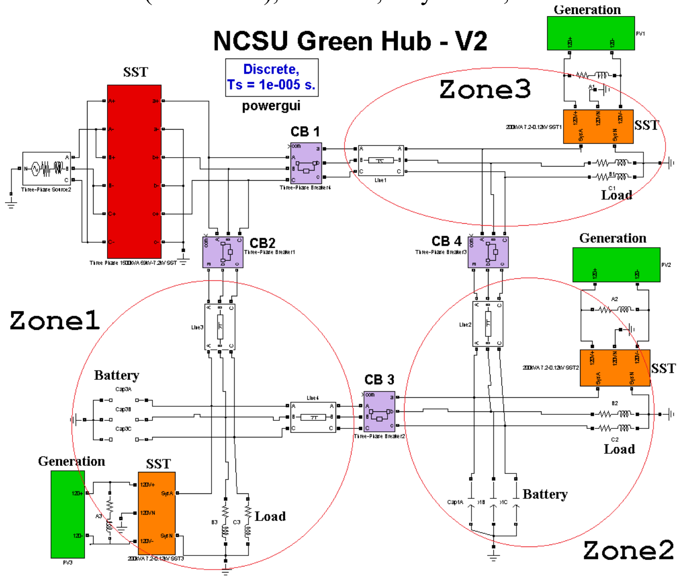

Fig. 1. FREEDM protection for GreenHub model. Division on zones.

Zone 0 protection is the differential protection described above, the current from zone 1, 2, 3 are evaluated at the central IFM, and in case of fault, trip is sent to corresponding FIDs.

FREEDM centre had prototyped this algorithm using LabView [3].

B. Load Balancing Application

Another DGI application critical for FREEDM operation is Load Balancing. Load balancing algorithm is used to normalize the overall load within FREEDM distribution network among the IEM peers. The DGI nodes, running the load balancing algorithm, communicate their load changes with each other in order to transfer the load from a node with excess load to a node that is need of additional load in order to maintain stability of the distribution system. The load balancing algorithm is developed by FREEDM centre and presented in [4]. This paper presents the implementation of the algorithm in a distributed manner using IEC 61499 function blocks technology. IEC 61499 enables modularity of the code, easy re-configurability of the developed system and results in executable system specification. The developed modular code is directly deployable to the hardware platform such as SST controller - ARM board.

Fig. 2 shows the state flow diagram of the distributed load balancing algorithm as presented in [4].

The simplified version of the algorithm is explained below in three simple steps:

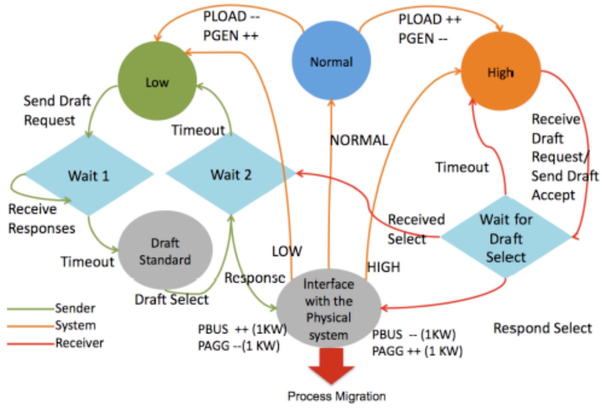

Fig. 2. State diagram of a distributed load balancing scheme [4].

1) Every IEM peer (DGI node) computes the actual (house) load of the SST (represented by X_{Actual}). Actual load is defined as the difference between total load consumption (represented by X_{Load}) and total load (power) production by the energy resources (DRER) (represented by X_{DRER}), i.e $X_{Actual} = X_{Load} - X_{DRER}$;

2) An IEM node is considered to be in a *Low* state (excess load, i.e. *Supply*), if actual load is less than *Threshold*, i.e. $X_{Actual} < Threshold$. It is thought to be in a *High* state (inadequate load, i.e. *Demand*), if actual load exceeds the *Threshold*, i.e. $X_{Actual} > Threshold$. Threshold is defined based on an optimization heuristic. In any other cases, the IEM is in a *Normal* state (has just sufficient load);

3) The IEM node in a low state then communicates with the IEM node in a high state for power migration following the state chart as shown in Fig. 2.

Complete algorithm details can be found in [4].

III. IEC 61499 Implementation of Distributed Grid Intelligence Applications

A. Implementation of Intellgent Fault Manafement Application

The control algorithm consists of differential and overcurrent protection. IEC 61850 models these functions as PDIF and PIOC Logical Nodes [5]. In the protection scheme IFM sends trip signal to circuit breaker (CB), therefore the control system should have CB model. According to the standard, CB is modelled as XCBR LN. Thus PDIF LN or PIOC LN issues trip signal to XCBR LN.

The control system consists of FBs – LN from the developed LN library (PDIF, PIOC and XCBR). There is a direct relation (mapping) between equipment and automation functions used in the Green hub system and corresponding FBs (iLNs) in the control model: circuit breaker – iXCBR, receiver of digitalized current samples – UDPSocketServer, differential protection – iPDIF and overcurrent protection – iPIOC.

More detailed description of IFM implementation is presented in [6].

The advantages of the function blocks architecture are the intuitive graphical design and the ease of distribution. Having a full library of necessary standard iLN, it is easy to implement a given protection/control system. The protection system is implemented by dragging and dropping necessary iLN, following the topology of the system. There are three zones in the control model. Each zone has same control/protection scheme, therefore iLNs used in all zones for the same function are the instances of the same iLN. For example, iPDIF1 in zone 1 and iPDIF2 in zone 2 are the instances of same iPDIF FB. Thus the IEC 61499 architecture provides code re-usability. Fig. 3 presents FB control model. In the right top corner the Matlab model of Green hub is presented, which was provided by FREEDM centre project partners. Current is measured at the given points and transmitted to FB model through UDP sockets. There is one FB to one UDP socket.

The processed current is fed to differential protection unit within the IFM – iPDIF. iPDIF, estimates the given samples, and in case of fault decides to initiate Operate (Trip) signal to circuit breaker – iXCBR. Since two zones share border CB, therefore trip signal goes to "OR" FB first (Fig. 3). XCBR sends trip signal through UDP to Simulink model of circuit breaker.

Each IFM node comprising of iPDIF and iPIOC - is placed within each zone. They are agents performing local decision making and contributing into DGI protection functionality.

Each function is implemented as a module, so the design is tidy, easy to follow, maintain, re-use and re-configure the system. Fig. 3 demonstrates a system level design of protection system for given Green hub model.

At this stage the design is purely hardware independent. The implementation of the protection scheme can reside in a single IED/PLC/PC (i.e. physical device) or be distributed across four devices according to the topology of the grid. In this case the communication between the physical devices is thought to use IEC 61850 protocol. Trip signals from control device to primary equipment circuit breaker are sent by means of GOOSE messages and current samples from merging units to IFM control device using the "sampled values" concept [7]. The communication between control devices can utilize client-server protocol defined in IEC 61850 [7].

Fig. 4 shows distribution of the DGI control logic over physical devices.

B. Implementation of Load Blancing Application

The Load Balancing application consists of 3 agents - IEM nodes, in *Low, High* and *Normal* states respectively [4]. Originally the system was modelled as multi-threaded system developed in C++ and running on a Linux machine. The code is not easily readable. The code must be modified for a target platform for deployment.

The IEC 61499 architecture allows for modular design of the distributed system, reflecting topology of the system. Each IEM node was designed as a Composite Function Block. In this work 2 node system was implemented as an example. As Fig. 5 shows the system consists of 2 FBs - 2 IEM nodes, and UDP FBs to interface Matlab model of the FREEDM system. To scale the system up to 3 nodes, the third FB IEM node needs to be instantiated from the developed library.

The IEM node is modelled as a composite FB - LBNode, presented on Fig. 6. On the right hand side of the picture there is the interface of the FB and on the left is its internals. The main components of the IEM node FB are "LoadBalanceNode" and "LoadTable". The rest FBs are communication blocks and auxiliary FBs. The LoadTable FB maintains the table of the IEM nodes and their current load. The LoadBalanceNode FB is

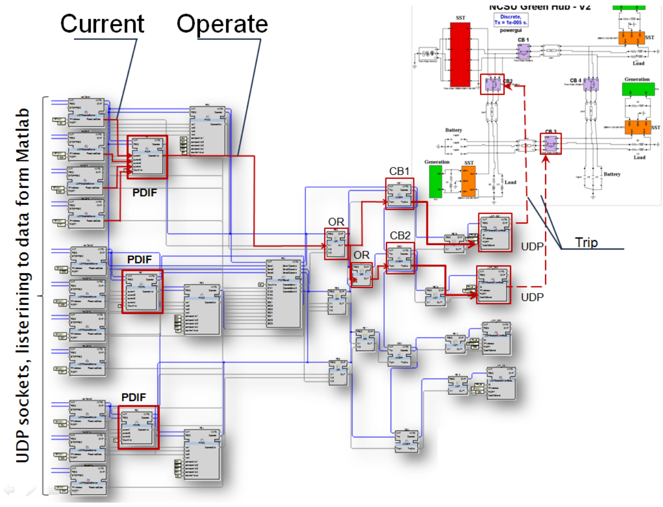

Fig. 3. System level design of the FREEDM protection scheme.

the basic FB executing the load balancing algorithm. Fig. 7 demonstrates its Execution Control Chart (ECC).

It is similar to the state machine in Fig. 2. The logic is briefly described below. At the predefined interval, the plant (in our case, Matlab simulation model) sends its current load readings to the controller in the IEM node. The controller then calculates the state of the node (*UpdateStateTable*) and then transitions to appropriate state (either *Supply* or *Demand* or remains in the same state if neither) and emitting a *NODE_STATE_UPDATE* event to other nodes to update the state in the state table of individual nodes. If it goes to *SUPPLY* state, it sends *DRAFT_REQ* message to all the nodes that in the *DEMAND* state. The *Supply* node then waits for *ACCEPT_DRAFT_REQ* response from the *Demand* nodes. If case of multiple *ACCEPT_DRAFT_REQ,* the first one to get a request from is considered in the simple implementation of this algorithm. In future implementations the selection will be performed by evaluation of factors like its own predicted need, economics and other optimization metrics [4]. On receiving *ACCEPT_DRAFT_REQ,* it transitions to *DraftStandard* state and sends out a *DRAFT_SELECT* event to the node from which it received the *ACCEPT_DRAFT_REQ.* At this point, the power transfer begins. In case the node enters the *DEMAND* state, it waits for the *DRAFT_REQ* event from a node which is in *SUPPLY* state; one receiving it sends *ACCEPT_DRAFT_REQ* and waits for *DRAFT_SELECT* in the *WaitDraftSelect* state. One receiving *DRAFT_SELECT*, It starts consuming load from the shared BUS.

The developed FB network of Load Balancing application of 2 IEM nodes is directly executable. Since SST controller is realized on ARM board, the FB network of Load Balancing application was deployed to the ARM board. There is no overhead of re-writing the code base to suit the target platform. The results of testing via co-simulation is described in the following section.

IV. RESULTS AND DISCUSSION

Implemented control system is validated using developed co-simulation environment. Fig. 8 illustrates the concept of the simulation method.

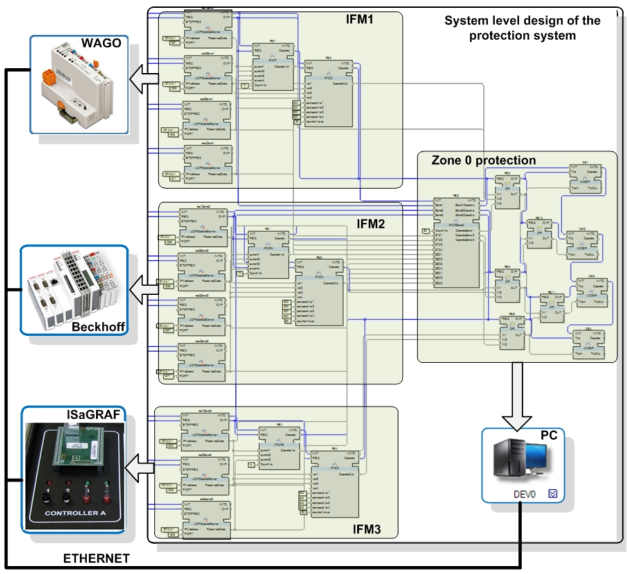

Fig. 4. Control system distribution, mapping to available hardware.

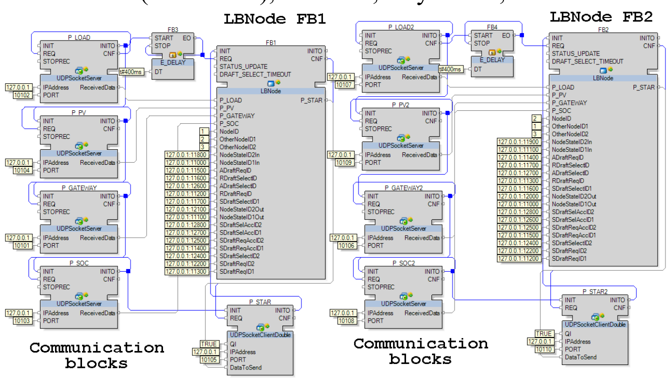

Fig. 5. Load Balancing application: two DGI nodes and UDP FBs to interface Matlab.

A. *Intelligent Fault Management application test*

The given Matlab model of Green hub system runs in parallel with implemented protection scheme for that system. Sampled current values from Matlab and trip commands from the FB control model are sent by UDP. This is the same approach as used in FLISR multi-agent system [8]. Function block control model is developed using nxtControl 1.4 development tool [9].

The fault is simulated to occur in zone 3, where IFM 3 is operating. All IFM agents are constantly monitoring current within assigned zones. IFM 3 will notice that the current is out of balance, when sum of the current samples is not equal to zero. It starts counting the number of consecutive instances where summed current is not zero. Once the number reaches pre-set value, in this case 10, IFM sends the trip signal to XCBRs 3 and 4, which will isolate the faulty zone by tripping (Fig. 9). Fault is simulated at approximately 0.18 seconds. IFM has isolated the fault at approximately 0.2 seconds. This is simulation time and does not correspond to real time due to computational overhead. This timing limitation of the simulation software Simulink is being investigated.

After the control logic has been validated, it was directly deployed to Beckhoff and WAGO PLCs. Both IEC 61499 and IEC 61850 ensure hardware independent design and interoperability between devices from different vendors.

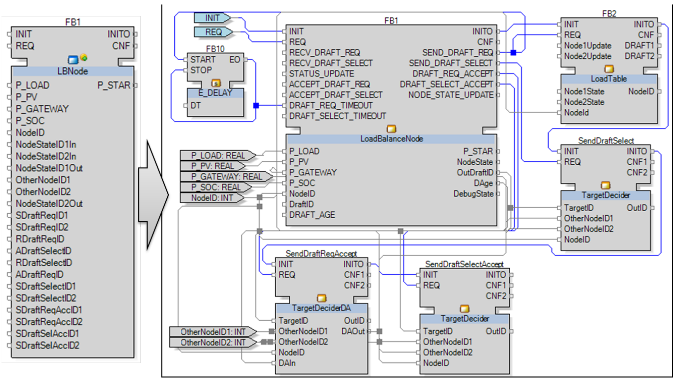

Fig. 6. LB Node and its composition

B. *Load Balancing application test*

Matlab model for this simulation is FREEDM System model with three SST nodes and it is the same model used to test original Power Balancing algorithm. More detailed description of the Matlab model can be found in [4]. Our example considers only two nodes.

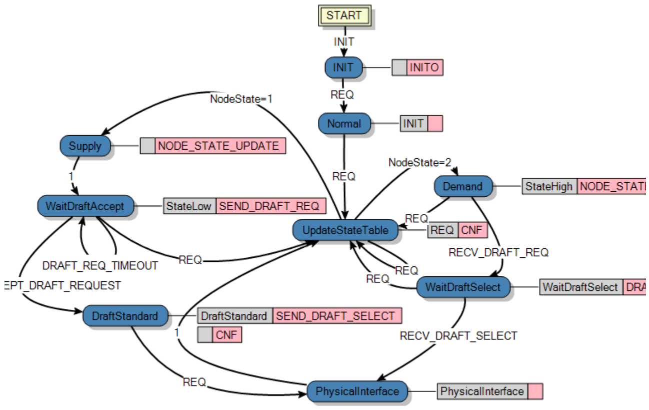

Fig. 7. ECC of the "LoadBalanceNode".

The Fig. 5 demonstrates system level FB network of 2 IEM nodes. The system was tested in a distributed set-up: node 1 is deployed to an ARM board, and node 2 is running on a PC. Two instances of FB type "LBNode" was deployed on a PC and ARM board, without modifying the code base.

The same concept of co-simulation described in previous subsection is used here. Matlab model of FREEDM system is running in parallel with the load balancing application running on distributed hardware.

Node 1 has no DRER and DESD; node 2 has both. Both nodes are in NORMAL state initially. The energy migration is shown in Fig. 11. P gateway is the power coming from the main grid. P* is the power dispatched by the SST with command of DGI node.

The node 2 has gone to state LOW at 0.310, since its PV generation produces excess power. And since the local node (node 2) was in state NORMAL, the battery is being charged from the generated power. Meanwhile, the load of node 1 has increased at 0.520s, and the node moved to state HIGH. Since the node 1 is in the HIGH state it will broadcast the new state and wait until a node in state LOW will send message *DRAFT_REQ*. Node 2 is in state LOW, and it will start power migrations to node 1 at 0.580s. As the increased power demand at node 1 balances out with the increased power supply from node 2, the power migration terminates at 1.220s.

V. Discussion on Distributed SCADA Concept

SCADA system is a central point of control and monitoring of any distribution system. With the distributed and dynamic nature of FREEDM network, traditional SCADA system needs to be changed. The events of plugging and unplugging DRER, DESD and PEV; and events of DGI nodes coming on- and off-line will need to be registered with the SCADA. So that HMI of the control room needs to be updated.

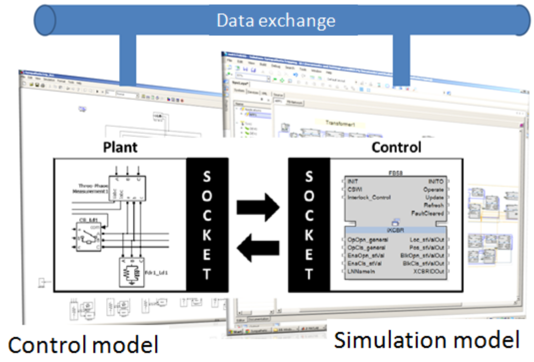

Fig. 8. Developed co-simulation environment.

Traditional SCADA extensively relies on pre-configured rigid system architecture and communication network topology. Such system does not have a strategy to handle dynamic re-configuration. Alternative solution might be distributed SCADA concept.

In this vision, distributed devices involve in assembling the central HMI, providing information about themselves to the SCADA master. Since all devices within FREEDM system carry some sort of "intelligence", they will notify SCADA HMI when they connect or disconnect to or from the network. SCADA HMI in turn will maintain up-to-date system topology and data. This concept requires that each component of the system is capable of self-description. Self-description of the device includes visual component and functional: how to communicate to the device, what services it offers and how to control it.

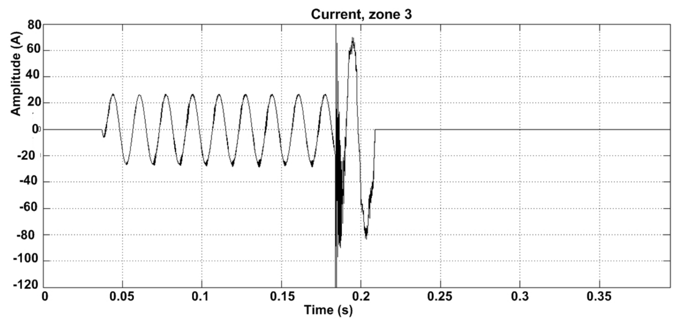

Fig. 9. Current on the load in zone 3.

On the SCADA master side, the HMI component of the SCADA is managed by special type of function block known as the CAT (Composite Automated Type) function block. CAT FB is not part of the IEC 61499 reference architecture. CAT blocks were introduced by nxtControl [9] to encapsulate Model-View-Control (MVC) design patter within IEC 61499. The CAT function blocks consist of two components: functional and visual. Fig. 10 illustrates the architecture of the SCADA system with CAT function blocks. There can be a CAT function block residing on the HMI server for each DGI node. Each DGI CAT function block will be collaborating with each other to manage the HMI of the FREEDM system. The functional component of the CAT function block will be processing the incoming data from the corresponding DGI and the visual component of the CAT function block will be displaying the monitored data of the FREEDM system. The proposed HMI of the SCADA system with function blocks will operate in a distributed manner between several CAT function blocks, each controlling the HMI of its corresponding DGI device. Hence, although all the data from the DGI will still be stored on the SCADA centrally, the SCADA HMI will be

Please cite as: G. Zhabelova, S. Patil, C.-W. Yang, V. Vyatkin, "Implementation of Smart Grid Intelligence Applications using IEC 61499 Reference Architecture", IEEE Conference on Industrial Informatics (INDIN'13), Bochum, July 29-31, 2013

managed in a distributed manner by several function blocks to form a single HMI overview of the entire FREEDM system.

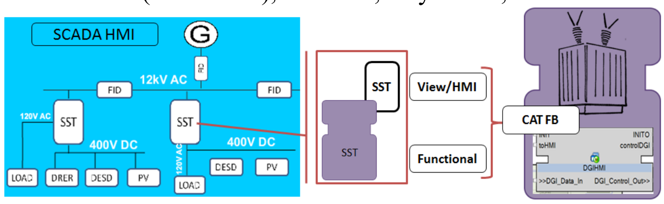

Fig. 10. FREEDM SCADA Architecture with Function Blocks.

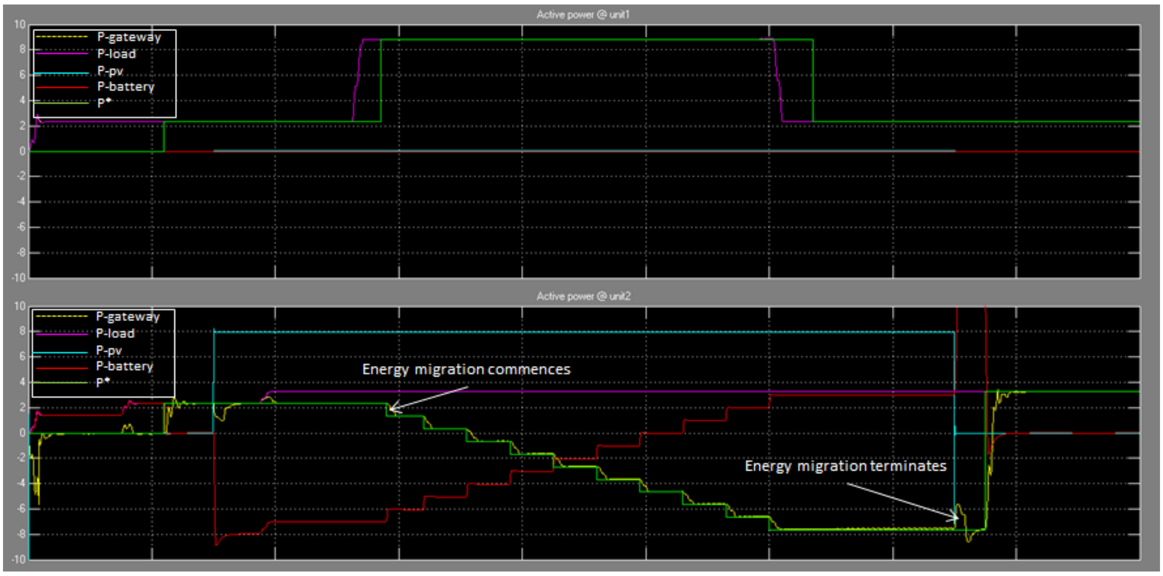

Fig. 11. Energy Migration as seen in Matlab

CONCLUSION

This paper presents IEC 61499 implementation of DGI applications. The approach is based on two standards IEC 61499 and IEC 61850. Combination of these can bring together benefits of distributed systems and unified communication protocol. This approach enables power system with multi-agent technology and its advantages. Developed IFM and IEM applications are directly executable on available hardware compliant with IEC 61499. There is no overhead of re-implementing validated algorithm to adapt it to the target hardware. Practical result was achieved with Load Balancing application which was deployed to ARM board (SST controller) and co-simulated with Matlab FREEDM system model.

The authors also discuss distributed SCADA concept. The proposal is to use IEC 61499 capabilities and CAT concept to model functional and visual components of the SCADA HMI.

Immediate future work is to implement service interface function blocks (SIFB) to access and use developed DGI services such as State Collection, Broker and Group Management. SIFBs will represent APIs, providing comprehensive access layer to the services. This will enable an engineer to develop DGI applications using IEC 61499 and re-use developed DGI services.

ACKNOWLEDGMENT

This work has been supported, in part, by FREEDM NSF Centre grant Y3.E.C12 and by the Uniservices, c/n 30328.

LIST OF ACRONYMS

AMI	Advanced Metering Infrastructure
CB	Circuit Breaker
DESD	Distributed Energy Storage Device
DGI	Distributed Grid Intelligence
DRER	Distributed Renewable Energy Resource
FB	Function Blok
FID	Fault Isolation Device
FREEDM	Future Renewable Electrical Energy Delivery and Management
HMI	Human Machine Interface
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEM	Intelligent Energy Management
IFM	Intelligent Fault Management
iLN	Intelligent Logical Node
LB	Load Balancing
LN	Logical Node
PC	Personal Computer
PDIF	Protection DIFFerential
PEV	Plugin Electric Vehicle
PIOC	Protection Instantaneous Over Current
PLC	Programmable Logic Controller
RES	Renewable Energy Resource
SCADA	Supervisory Control And Data Acquisition
SIFB	Service Interface Function Block
SST	Solid State Transformer
XCBR	X-Primary Equipment, Circuit BReaker

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] "Function blocks IEC 61499 - Part1: Architecture," ed: 2, International Electrotechnical Commission, 2012.

[3] A. Thirumalai, X. Liu, and G. Karady, "Novel Digital Protection System for FREEDM Loop," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, 2010, pp. 22-26.

[4] R. Akella, M. Fanjun, D. Ditch, B. McMillin, and M. Crow, "Distributed Power Balancing for the FREEDM System," in *Smart Grid*

Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 7-12.

[5] International Electrotechnical Commission, " Basic information and communication structure," in *Communication Networks and Systems in Substations*, ed. Switzerland: International Electrotechnical Commission,, 2010.

[6] G. Zhabelova, V. Vyatkin, and N. Nair, "Standards- based Intelligent Fault Management System for FREEDM Green Hub Model," in *IEEE International Conference on Industrial Electronics (IECON’11)*, Melbourne, Australia, 2011.

[7] International Electrotechnical Commission, " Basic information and communication structure - Abstract Communication service interface (ACSI)," in *Communication Networks and Systems in Substations*, ed. Switzerland: International Electrotechnical Commission,, 2010.

[8] V. Vyatkin, G. Zhabelova, N. Higgins, M. Ulieru, K. Schwarz, and N. K. C. Nair, "Standards-enabled Smart Grid for the future Energy Web," in *Innovative Smart Grid Technologies (ISGT)*, Gaithersburg, MD, 2010, pp. 1-9.

[9] GmbH nxtContol, "nxtStudio ", 1.4 ed, 2010.