Fault Detection, Isolation and Restoration Test Platform Based on Smart Grid Architecture Model Using Internet-of-Things Approaches

Abouzar Estebsari
Department of Energy
Politecnico di Torino
Turin, Italy
abouzar.estebsari@polito.it

Edoardo Patti and Luca Barbierato
Department of Control and Computer Engineering
Politecnico di Torino
Turin, Italy
edoardo.patti@polito.it, luca.barbierato@polito.it

Abstract—**To systematically shift existing distribution outage management paradigms to smart and more efficient schemes, we need to have an architectural overview of Smart Grids to reuse the assets as much as possible. Smart Grid Architecture Model offers a support to design such emerging use cases by representing interoperability aspects among component, function, communication, information, and business layers. To allow this kind of interoperability analysis for design and implementation of Fault Detection, Isolation and Restoration function in outage management systems, we develop an Internet-of-Things-based platform to perform real time co-simulations. Physical components of the grid are modeled in Opal-RT real time simulator, an automated Fault Detection, Isolation and Restoration algorithm is developed in MATLAB and an MQTT communication has been adopted. A 2-feeder MV network with a normally open switch for reconfiguration is modeled to realize the performance of the developed co-simulation platform.**

Keywords—*Fault detection; Smart Grid Architecture Model; restoration; real time simulation; Internet-of-Things.*

I. INTRODUCTION

In order to systematically shift existing control and protection paradigms to smart grids, we need to map new use cases to Smart Grid Architecture Model (SGAM). SGAM offers a support for design of new use cases in smart grids with an architecture approach by which interoperability viewpoints can be well represented [1].

In interoperability analysis of smart grids, following questions are addressed: i) What are physical components required? ii) How should the information be exchanged? iii) What data should be communicated? iv) What functions are needed? v) What are business and regulatory constraints to be applied? These considerations respectively form 5 smart grid interoperability layers including *Component layer*, *Communication layer*, *Information layer*, *Function layer* and *Business layer*, respectively (see Fig. 1). *Component layer* accommodates the physical distribution of all participating components in the smart grids context including hardware devices, physical infrastructure, entities and all actors. *Communication layer* addresses protocols and mechanisms for the interoperable exchange of information among actors.

Information layer contains measurements, alarms, commands and in general all exchanged data. *Function layer* represents smart services in an advanced distribution management system (e.g. advanced outage management, demand response, distributed energy resources control). Finally, business objectives, political and regulatory framework to develop a use case should be mapped into the *Business layer*.

Similar to any other ex-ante analysis, we need laboratory facilities and simulation systems to test and verify performance of new methods or tools for different use cases. As a crucial requirement for test and validation of new solutions, a near real-world environment is recommended to be used [2]. In this regard, real-time simulations (RTS) depict a path that can effectively support the research efforts to meet emerging lab test requirements [3 - 5]. For power system electromagnetic transient analysis (EMT), like fault analysis, real time simulation is very reliable with flexibility and scalability due to its computation parallelism feature. A review of offline and real-time simulations for EMT is reported in [6].

However, to study interoperability in smart grids based on SGAM, combination of control and communication emulators with grid real-time simulator is needed. In other words, a co-simulation framework is required to concurrently reproduce the behavior of different interoperability layers of the specific use case. Considering the advantages of RTS, this implies distributed real-time co-simulation over networked control systems. An overview of existing co-simulation frameworks is discussed in [7]. Some of these frameworks do not run simulations in real-time, some are not scalable in terms of number of interconnected devices (e.g. smart meters, relays, actuators, etc.) and some are quite costly in terms of lab set-up.

Our contribution is to develop a co-simulation framework using Internet-of-Things (IoT) communication paradigms and protocols along with digital real-time simulators (DRTS) to create a flexible and scalable Software-In-the-Loop and Hardware-In-the-Loop platform for outage management in smart grids. The IoT approach allows interconnecting many virtual or physical devices [8, 9] (i.e. circuit breakers, reclosers, remote terminal units, smart meters, etc.) and/or several simulation modules [15].
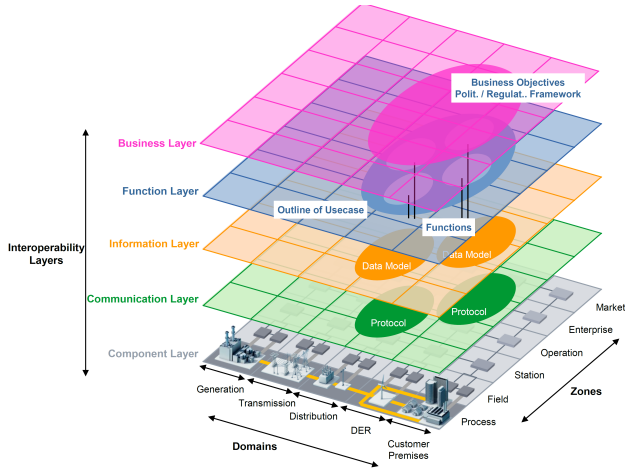
Fig. 1 Smart grid Architecture Model with Interoperability Layers

Besides establishing interfaces for interconnecting devices and simulation modules of this framework, a simplified emulator of the communication network needed for centralized Fault Detection, Isolation and Restoration (FDIR) is developed, which emulates the real communication delay considering a relevant protocol. The communication system simulator calculates the delay based on several factors from the type of involved Intelligent Electronic Devices (IEDs), exchanged information, traffic amount, function and protocols. This is very important in reliability analysis of distribution networks. This co-simulation would allow for assessment of the feasibility and performance of new FDIR algorithms with different communication protocols and information rate in smart grid, to understand how the reliability indexes of the system would improve.

The rest of this paper includes an introduction to the developed framework in Section II, implementation of this platform in our lab with some test results in Section III, and conclusion with remarks.

II. DEVELOPED FRAMEWORK

Our developed architecture enables testing different protection schemes and fault scenarios, safely, in a plug-and-play fashion with minimum amount of efforts thanks to its flexibility and scalability features. This architecture contains 3 main blocks: real time simulator (for power system component layer modelling), smart function host server (in the case study, centralized FDIR algorithm), and a Message Broker that implements the Message Queueing Telemetry Transport (MQTT) protocol [16]. The overall bidirectional structure of this framework is presented in Fig. 2.

Centralized FDIR [13] is a part of Outage Management System (OMS) in Distribution Management System (DMS). Advanced distribution system automation relies on 3 types of devices as Intelligent Electronic Devices (IED) (to perform functions like measurement, recording, control and protection), bay control, and Remote Terminal Units (RTU).
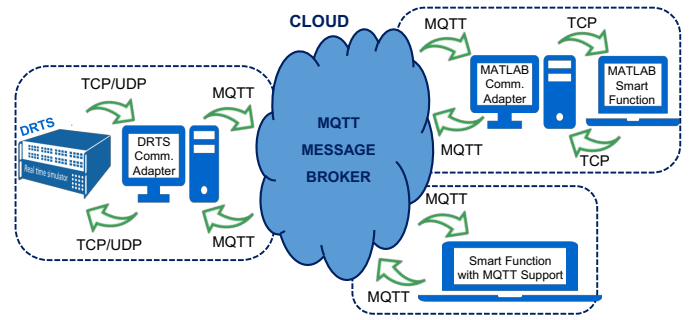

Fig. 2. Overall bidirectional architecture of the developed framework

Supervisor Control and Data Acquisition (SCADA) system of DMSs communicates with distributed RTUs in the network retrieving measurements and alarm signals, and sending control or protection commands. The implementation of simulations in a virtual environment makes testing of new control strategies easier, less expensive, and it enables a safe first validation of new projects without requiring costly implementation of fault scenarios in the real world.

The developed architecture enables testing different protection schemes and fault scenarios in a plug-and-play fashion with minimum amount of efforts thanks to its flexibility and scalability features.

To shortly discuss how our developed framework is suitable to study FDIR use case mapped in SGAM, we review some of SGAM interoperability layers corresponding to the elements of our framework: physical components including distribution grid, IEDs, RTUs and actuators (e.g. reclosers, reconfiguration switches, and circuit breakers) are all modeled to run in real-time using a digital real-time simulator (DRTS). DRTS also allows for integrating some real devices through performing Hardware-In-the-Loop (HIL) experiments. This represents the *Component Layer*.

OMS actions including FDIR [11], fault location [12], outage detection, outage mapping, and event analysis and recording can concurrently run on different hardware or software platforms, similar to the real world where they are hosted either in SCADA/DMS centralized system or in substations [13] (decentralized). This would realistically simulate the *Function Layer*.

Data exchanged among functions and physical components can be controlled to follow a specific protocol or media through a communication network emulator. Although for demonstration purpose, we limited the functionality of our communication emulator to delay generation, but this module can be easily replaced with an advanced communication network simulator. One can study different solutions for communication layer of FDIR thanks to this integration. Data provisioning and information monitoring in terms of type and size is also managed in our developed framework to address the *Information layer*. In other words, all exchanged information as well as sender and receivers can be easily extracted and indicated using our architecture. In FDIR, in the information layer, we mainly deal with alarms retrieved from the grid and commands generated by the function.

On the left side of the architecture shown in Fig. 2, there is a DRTS which accommodates the model of the distribution network with all physical components and devices. DRTS is capable to be connected to real devices (e.g. relays) through HIL. In our platform, an Opal-RT simulator is used with eMEGAsim configuration which is suitable for EMT analysis. From the data exchange point of view, DRTS is able to establish TCP or UDP communication with other devices and/or computers in the infrastructure.

The *DRTS Communication Adapter* next to the DRTS is executed on a different computer. It is able to receive from and send data to the simulator according to the chosen communication protocol (TCP or UDP). On the other side, it also interfaces with the *MQTT message broker*. The communication is based on a publish/subscribe paradigm [17]; it forwards data by publishing them and receives commands for the simulator by subscribing to them. Thus, the *DRTS Communication Adapter* acts as a bridge to provide DRTS with MQTT functionalities, both for publishing and subscribing.

The actors in the proposed architecture exploit the JSON open-standard format to exchange information among them. We choose JSON because it is becoming a common data-format for data exchange in the Web and across IoT devices. It uses human-readable text to transmit data objects consisting of key-value pairs. On the other side, the DRTS's data-format consists of a vector of numbers that we organized also as a series of key-value pairs where odd positions are our numerical keys and even positions are the numerical values. Thus, the *DRTS Communication Adapter* works also as a translator: it translates the numerical key-value pairs in the vector into key-value pairs in the JSON and vice-versa.

In the middle, there is the MQTT broker, an entity in charge of handling MQTT clients and forwarding published messages to the subscribers. This broker has a bidirectional operation.

On the top right side of the MQTT message broker in Fig. 2, there is a second *Communication Adapter* that is needed to provide smart functions with MQTT functionalities, both for publishing and subscribing. In this architecture, smart functions are developed in Matlab, which does not provide libraries to establish an MQTT communication. Thus, this *Matlab Communication Adapter* performs a crucial role to enable a bidirectional communication with the other actors in the architecture. It forwards data to the smart function host and sends back to the rest of the architecture the commands resulting from the smart function execution. It interacts through the MQTT protocol on one side, and it behaves as a TCP client towards the MATLAB smart function host. Unlike the *DRTS Communication Adapter*, this adapter does not translate the JSON since smart functions are ready to receive and post-process this data-format. This *Matlab Communication Adapter* is not needed if smart functions are developed in other programming languages that already support the MQTT protocol (e.g. C++, java, python, etc.).

Smart function (i.e. FDIR) host works as a TCP server, and it can be executed on a different terminal device, too (Fig. 3). Once it receives data from the rest of the architecture, it processes them, and sends commands back to the grid model in DRTS to manage the status of the grid. In our case study, an algorithm for FDIR is the smart function, however the platform is flexible to integrate more than one smart function and realize distributed intelligence.

Since this architecture has been designed to be modular, smart functions can be replaced in a plug-and-play fashion. Thanks to the MQTT protocol, adopted to exchange data across the actors in the architecture, also the DRTS can be replaced with real Internet-of-Things devices deployed across the distribution network. Thus, the architecture is able to switch from test-bed environments to real-world scenarios. These replacements or updates can be done without affecting the rest of the architecture.

III. IMPLEMENTATION AND RESULTS

FDIR aims to automatically provide detection and isolation of the faulted portion of distribution system and restore the service for customers on healthy sections of the grid. This requires wide deployment of intelligent devices and actuators in the field from one side and using an efficient communication system with low latency.

The conventional outage management systems are based on trouble calls, made by customers, for the outage mapping, and dispatching repair crew to isolate the fault and restore the rest of feeder. Comparing the conventional procedure with FDIR highlights the advantage of applying FDIR for system reliability improvement by reducing interruption time and also even decreasing the number of recorded permanent faults if the whole restoration process takes less than the thresholds for interruption registration in DSO.

To implement FDIR use case based on the described framework in previous section, a laboratory set-up shown in Fig. 3 is prepared using OP5600 Opal-RT simulator for running grid model. A PC as the smart function host runs FDIR algorithm on MATLAB, while MQTT message broker and adapters are running on another computer.

The 2-feeder MV distribution network depicted in Fig. 4. is used as a case study. This network is extracted from a portion of an urban distribution system in Northwest of Italy. Physical behavior of this electricity system is simulated in an Opal-RT DRTS by modelling network elements (e.g. transformers, branches, secondary substations with LV grid equivalent model, etc.) and protection functions (e.g. circuit breakers, relays, remote terminal units (RTU), etc.).
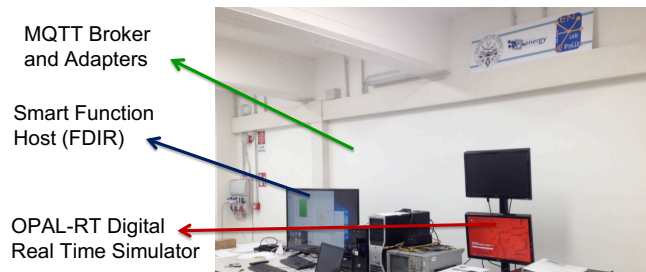

Fig. 3. Laboratory set-up of developed co-simulation architecture
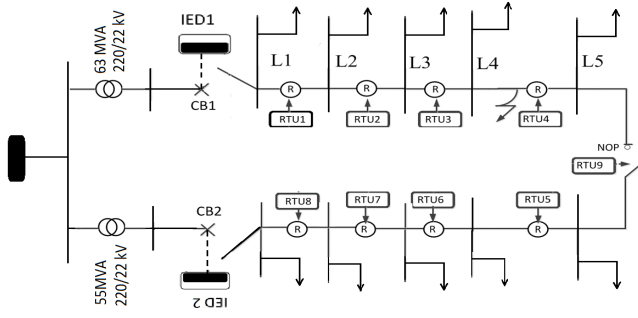
Fig. 4. MV 2-feeder distribution network used for the case study

When a fault occurs, an overcurrent protection relay at the beginning of MV feeder senses the violation and trips the Circuit Breaker (CB). At this point an Intelligent Electronic Device (IED) communicates this change to the FDIR centralized control server (i.e. FDIR agent).

FDIR agent will consequently send a command to all RTUs connected to the reclosers of that feeder up to the Normally Open Switch (NOP) in order to open them. The NOP is normally open to operate the network radially and will be used for network reconfiguration.

Once the reclosers are open, the FDIR agent commands to the IED to close the circuit breaker and then, starting from the first recloser after CB, the FDIR agent tries to close reclosers one by one through sending commands to corresponding RTUs. Once closing a recloser results in CB reaction, FDIR agent receives fault alarm from IED. Hence FDIR agent "detects fault" and leaves that recloser open as well as the next one located downstream. It then closes all the rest of reclosers up to the NOP.

In this way, the faulty portion of the feeder remains "isolated". Then, FDIR agent sends commands to the rest of open reclosers and NOP to re-energize the rest of the feeder by supplying them from a new feeder.

The aforementioned process is examined for a fault occurrence after aggregated load L4. The final snapshot of the network status is depicted in Fig. 5.
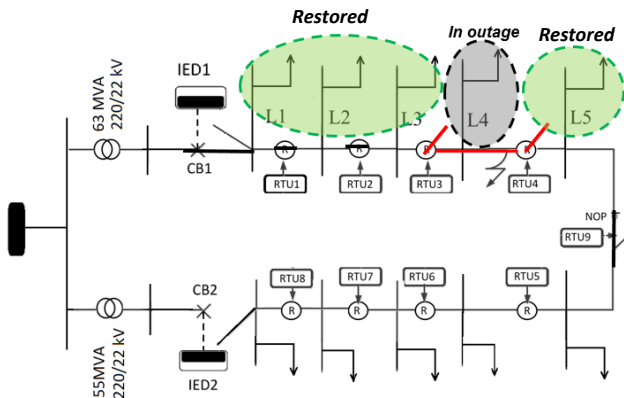

Fig. 5 Final snapshot of network status after FDIR actions

FDIR is updated of the network configuration. All commands and alarms (and in case measurements) data include also device IDs so that FDIR agent can associate them correspondingly. In our implementation, both FDIR server and DRTS are in the same LAN. Thus, the delay introduced by the communication architecture can be neglected. However, in real world, RTUs are distributed in a wide geographic area (e.g. a city). Among wired and wireless media to support this communication, we use our developed platform to make a comparison among mobile networks (2G, 3G and 4G technologies) for FDIR support. The idea is to realize the applicability and advantage of this simple platform to perform such ex-ante performance analyses.

Typical end-to-end delays of 2G, 3G, and 4G are 500-1000, 100-500, and less than 100 milliseconds respectively [14]. Data rates of 2G, 3G, and 4G is also reported as 100-400 Kbps, 0.5-5 Mbps and 1-50 Mbps, respectively [14]. The main component affecting the above-mentioned delay is related to the Internet routing latency, which depends on the instantaneous condition of the network. In our case study, we perform the worst-case analysis. We assume the FDIR agent is in Distribution Management System (DMS) located at 200 km far from the field. In this case, RTUs communicate with FDIR through SCADA. Considering the light speed, the propagation delay is computed as 200 km divided by 2×10^8 m/s, which is equal to 1 ms. Instead, the transmission delay which is defined as the ratio of packet length and data rate does depend on access technology and amount of transmitted data. In our case study, dimension of packet sent is about 66 Bytes (as seen through a network analysis on Wireshark), so even by using the 2G technology, which provides the lowest bandwidth, the transmission delay is in the order of few milliseconds. Therefore, both propagation and transmission delay can be neglected.

In our case study, each recloser receives signals 2 times from the agent. Considering maximum 1 second computation time for FDIR Matlab function, we add this 1 second as a delay to actuation of reclosers. In addition, all these reclosers as well as the circuit breaker and NOP communicate signals 2 times each one with network delay. So, the total delay is estimated as equation (1):

$$D_t = 2 \times (D_a \times N_r + D_n \times (N_r + 2)) \qquad (1)$$

Where D_t is the total delay, D_a is the FDIR agent delay, N_r is the number of reclosers in the feeder and D_n is the network delay. While the first term is related to FDIR agent decision making process (e.g. MATLAB function in our case), the second term depends on the chosen communication medium.

In order to improve electricity distribution reliability, 2 indexes namely System Average Interruption Duration Index (SAIDI) for each customer served and System Average Interruption Frequency Index (SAIFI) should be improved. The former is already considered once FDIR is applied instead of conventional trouble call based methods. The latter depends on the number of registered long-term faults. In different regulations and countries, from 1 to 3 minutes are the thresholds between short-term and long-term fault.
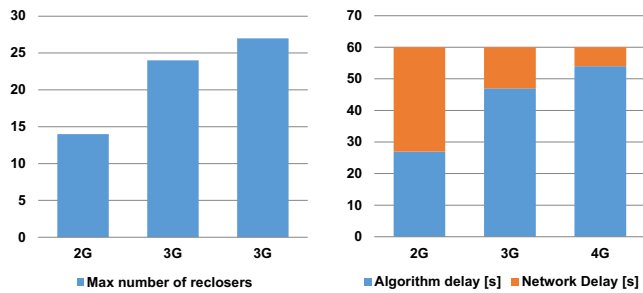
Fig. 6. Maximum number of reclosers (left) and delay contribution (right) for different mobile technologies

In our case, we assume 1 minute as the maximum accepted delay for which the fault occurrence is not recorded. In this case, according to the total delay calculation, no more than 30 reclosers should exist in one feeder. Obviously, as we scale up towards more recent technologies we are able to serve a higher number of reclosers and the contribution to the delay due to the algorithm becomes much more relevant (Fig. 6).

From Fig. 6, we can see that with 2G the two contributions (i.e. algorithm delay and network delay) are almost equal, while with 4G the algorithm delay is about 95% of the total one. So, in order to reduce the overall delay, we should try to improve the algorithm rather than scaling up with the communication technology.

IV. CONCLUSION

In the context of SGAM, we developed a real-time co-simulation platform for distribution system use cases in particular with the following characteristics: i) many devices distributed widely in the network, ii) phenomenon with fast transients like faults and iii) functions requiring fast reaction and communication like FDIR.

We exploited an IoT-based approach to create a flexible and scalable architecture suitable for these kinds of use cases. A case study with a 2-feeder network affected by a single-phase to ground fault, mobile communication network and FDIR algorithm is demonstrated. This tool can support validation of new protection schemes to improve reliability indexes. Different communication protocols to speed up the restoration process can be as well examined.

REFERENCES

[1] CEN-CENELEC-ETSI Smart Grid Coordination Group, Smart Grid Reference Architecture. November 2012.

[2] H. Farhangi, "The path of the smart grid," IEEE Power Energy Mag. 2010, 8, 18–28.

[3] R. Podmore and M.R. Robinson, "The role of simulators for smart grid development," IEEE Trans. Smart Grid, 2010, 1, 205–212.

[4] C. Dufour and J. Belanger, "On the use of real-time simulation technology in smart grid research and development," IEEE Trans. Ind. Appl. 2014, 50, 3963–3970.

[5] F. Guo, L. Herrera, R. Murawski, E. Inoa, C.L. Wang, P. Beauchamp, E. Ekici, and J. Wang, "Comprehensive real-time simulation of the smart grid," IEEE Trans. Ind. Appl. 2013, 49, 899–908.

[6] J. Mahseredjian, V. Dinavahi, and J. A. Martinez, "Simulation tools for electromagnetic transients in power systems: overview and challenges," IEEE Trans. Power Deliv. 2009, 24, 1657–1669.

[7] W. Li, X. Zhang, and H. Li, "Co-simulation platforms for co-design of networked control systems: An overview," Control Eng. Pract. 2014.

[8] E. Patti and A. Acquaviva, "IoT platform for Smart Cities: Requirements and implementation case studies," 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), September 2016.

[9] I. Khajenasiri, A. Estebsari, M. Verhelst and G. Gielen, "A Review on Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications," Energy Procedia, 2017.

[10] A. Estebsari, A. Bahmanyar, E. Pons, E. Patti, M. Mengistu, S. Jamali and E. Bompard, "An IoT realization in an interdepartmental real time simulation lab for distribution system control and management studies," 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, 2016, pp. 1-6.

[11] A. Zidan et al., "Fault Detection, Isolation, and Service Restoration in Distribution Systems: State-of-the-Art and Future Trends," in IEEE Transactions on Smart Grid, vol. 8, no. 5, pp. 2170-2185, Sept. 2017.

[12] A. Estebsari, E. Pons, E. Bompard, A. Bahmanyar and S. Jamali, "An improved fault location method for distribution networks exploiting emerging LV smart meters," 2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), Bari, 2016, pp. 1-6.

[13] P. Parikh, I. Voloh and M. Mahony, "Distributed fault detection, isolation, and restoration (FDIR) technique for smart distribution system," 2013 66th Annual Conference for Protective Relay Engineers, College Station, TX, 2013, pp. 172-176.

[14] I. Grigorik, "High performance browser networking," O'Reilly, 2013.

[15] L. Bottaccioli, A. Estebsari, E. Pons, E. Bompard, E. Macii, E. Patti, A. Acquaviva, "A Flexible Distributed Infrastructure for Real-Time Cosimulations in Smart Grids," in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3265-3274, Dec. 2017.

[16] Message Queueing Telemetry Transport (MQTT), Available on-line: http://mqtt.org/

[17] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," ACM Comput. Surveys, vol. 35, pp. 114–131, Jun. 2003