AN ADAPTIVE, AGENT-BASED PROTECTION SCHEME FOR RADIAL DISTRIBUTION NETWORKS BASED ON IEC 61850 AND IEC 61499

Daniele PALA
RSE spa – Italy
daniele.pala@rse-web.it

Carlo TORNELLI
RSE spa – Italy
carlo.tornelli@rse-web.it

Gianluigi PROSERPIO
RSE spa - Italy
gianluigi.proserpio@rse-web.it

ABSTRACT

This paper shows a possible implementation for an auto configurable protection scheme based on the well-known logic selectivity concept, using an agent-based architecture with standard communication protocol (IEC 61850) and programming language (IEC 61499). An easy-to-use technique to integrate the two standards is shown, enabling the exploitation of encapsulation capabilities provided by IEC 61499 to provide easy to maintain and reusable code, along with interoperability with 61850-enabled IEDs. The proposed scheme can-also implement automatic reconfiguration of protections following network topology changes.

INTRODUCTION

Distributed generation causes several challenges to the protection of distribution networks, such as increased or decreased fault levels, false tripping of feeders and blinding of protections. With special reference to the first issue, selectivity problems arise, due to the possible contributions of distributed energy sources to fault currents. Moreover, the possibility of dynamic network reconfiguration (for optimization purposes, for example) should be taken into account when designing a network protection scheme. For these reasons adaptive protection and logic selectivity are good candidates for the implementation of protection schemes in radially-operated distribution networks with high DG penetration.
The use of agent-based technologies in this context may allow the splitting of the whole logic into smaller and self-contained building blocks, resulting in better scalability and software reuse and modification. Finally, the use of existing standard communication protocols and languages must be taken into account.
This paper shows a possible implementation of an auto configurable protection scheme based on the well-known logic selectivity concept, with the following characteristics:

- Agent-based architecture
- Full automatic reconfiguration following network topology changes, even in absence of faults.
- Use of standard communication protocol (IEC 61850) and programming language (IEC 61499), with an easy-to-use technique to integrate the two.
- Exploitation of encapsulation capabilities provided by IEC 61499 to provide easy to maintain and reusable code.
- Straightforward extensibility of code for additional functionalities, like anti-islanding protection.

The use of existing standard communication protocols and languages has thus been taken into account through the use of IEC 61850.
The topic of integration between IEC 61850 and IEC 61499 has been already addressed in the literature [1], [2], [3], however, for this application, we propose a simpler approach, by restricting our scope to a subset of the 61850 standard, namely GOOSE messaging. This makes the implemented solution more straightforward and with less overlap between the two standards: more specifically, we exploit the analogy between the 'event' concept as defined by IEC 61499 and GOOSE to obtain a very simple integration methodology.
The paper describes the software framework used for the development of adaptive protection logics which takes advantage of all the superior features of IEC 61499 as opposed to IEC 61131, while allowing an extremely easy integration with IEC 61850. Despite the simplicity of logic functions in this application, the design rules followed in the joint use of the two standards may be easily applied to more complex cases.

LOGIC SELECTIVITY

Logic selectivity is a well-known protections coordination technique which can be used in any radially-operated distribution network to achieve protections coordination. The principle of operation, applied to a MV feeder, is shown in Figure 1.

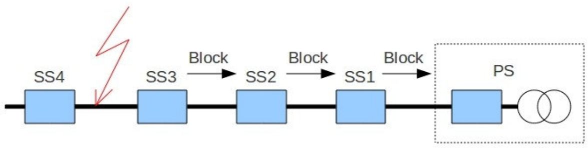

Figure 1 – Logic Selectivity

An automatic breaker installed in each secondary substation (SSx) opens when the current flowing from primary substation (PS) exceeds a configured maximum value, protecting the downstream section of the feeder. To simplify exposition, in this paper we identify a substation with a single breaker; the algorithm is unaffected by this simplification.
A fault located between secondary substation SS3 and SS4 is detected by all the protection devices ‘upstream’ that point, i.e. SS3, SS2, SS1 and PS (Primary Substation). Every device that senses the fault, immediately sends a blocking signal to its ‘predecessor’ to inhibit it, so that only the device which doesn’t receive the blocking signal within a specified timeout (SS3 in this case) will trip. After tripping, the breaker that opened sends a trip command to the downstream device (SS4 in this case) to complete the fault isolation procedure.
This algorithm has gained interest for the protection of

MV feeders as the availability of communication between devices has become more widespread, as well as the availability of intelligent fault current interrupting devices, suitable for installation inside secondary substations [5], [6].
In case of loss of the communication channel between protections, it is possible to use a 'local' protection logic as backup, like the one described in [5].

Handling reconfigurations

Despite its simplicity, logic selectivity is a good starting point for the implementation of agent-based protection logics, since it involves only local communication between neighboring nodes. An immediate problem arising from this scheme is network topology awareness: every node must always be sure to have the correct information about who is its 'predecessor' as well as his 'successor' along the feeder. Due to the tree structure of distribution networks, there are situations in which one node has more than two neighbors. This typically occurs in correspondence of branches of the network. Moreover, network topology may change in consequence to a fault or maintenance interventions, but also for optimization purposes (e.g. losses minimization). We'll show now that it is possible to obtain local awareness about network topology with a very simple and robust distributed algorithm.
For this purpose, every secondary substation breaker maintains two disjoint lists of neighbors, one for each of its terminals. Primary substations have a unique list of neighbors because only one terminal is considered.
For substation along a feeder, there is normally only one neighbor for each terminal. If the substation is near a branch of the network one terminal has more than one neighbor.

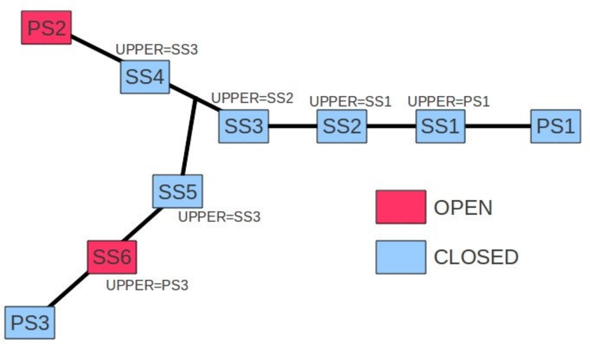

Figure 2 - Example network

With reference to the example network in Figure 2, node SS3 has SS2 as neighbor on his right terminal and SS4 and SS5 on his left.
Every secondary substation node in the network maintains a variable, called UPPER, which identifies its predecessor (i.e. the preceding node along the path to the feeding substation and to which the blocking signal must be sent in case of fault). Every neighbor which is not the UPPER is identified as a LOWER node. Primary substations have no UPPER node. The distributed algorithm automatically adapts the protections' intervention logic to the current network topology by updating the UPPER variable of each protection device. It works as follows: every node updates its configuration when it receives from its neighbors a message, that we call UPDATE. This message, when generated by a node, is sent to all its neighbors. Each open breaker in the network send periodically the UPDATE message to its neighbors.
The update procedure is then initiated by the open breakers (either in primary or secondary substations) and propagates along the feeder. The updating logic is described in Table 1.

	Breaker Opened	Breaker Closed
Primary Substation Node	Periodically send UPDATE message.	Send UPDATE message when received from all neighbors.
Secondary Substation Node	Periodically send UPDATE message.	This condition is where the actual updating of the UPPER parameter takes place (See Table 2).

Table 1 – Reconfiguration logic

The updating logic implemented inside each SS node is shown in Table 2.

Step 1: wait for UPDATE from neighbors.
Step 2: if UPDATE is received from all neighbors of one terminal list, send UPDATE and wait for UPDATE from the remaining neighbors.
Step 3: if UPDATE is received from all neighbors, set as UPPER node the last neighbor from which you received the UPDATE and go back to Step 1.

Table 2 – Algorithm: Secondary Substation, closed breaker

Since the UPDATE message is generated periodically by the open breakers, the procedure will converge to the correct configuration.
An example illustrating the sequence of UPDATE messages exchanged between SS4, SS5 and neighbor nodes to determine their predecessor (SS3 in this case) is shown in Figure 3.

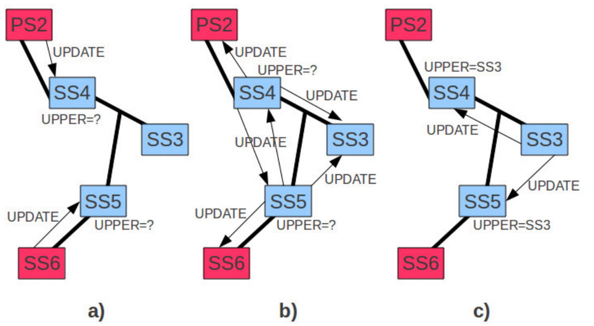

Figure 3 - Update sequence

In Figure 3 a) the configuration sequence starts from SS6 and PS2 that, being open, send the UPDATE messages to their neighbors (SS5 and SS4 respectively). In Figure 3 b) SS4 and SS5, according to Table 2 , having received the UPDATE message from all their neighbors connected to one terminal, execute Step 2 and thus send UPDATE messages to their neighbors. In Figure 3 c) is the node SS3,that sends UPDATE, having received UPDATE from all his neighbors (SS4 and SS5) on the same terminal. At this point SS4 and SS5 have received UPDATE command form all its neighbor and can execute Step3 of the algorithm. Since the last neighbor from which they received the UPDATE message is SS3, they set it as their UPPER node.

The open breakers work by sending UPDATE messages periodically; however, they may also work in an event-triggered fashion, without any message replication, but in this case the algorithm wouldn't be robust with respect to lost messages.

Note that every node is only concerned with information coming from its neighbors, it does not need knowledge about the full feeder topology.

This kind of updating mechanism, if required, can also be used to force disconnection of distributed generation from the grid in case of fault, to prevent islanded operation of a part of the network (anti-islanding protection) In this case a node must know the list of distributed generator located 'downstream' its position, and this information must be consistent with current network topology. The communication can thus be extended to include also other types of agents, like distributed generators. The exchange of information with DG, apart from anti-islanding information, can involve the actual status of the generators (the actual generated power, for example) to switch between different setting groups.

The updating mechanism for the protections configuration can easily adapt to many different grid topologies and doesn't need a priori knowledge of the full feeder topology. An important feature of this algorithm is its distributed nature, based on simple building blocks ('agents') exchanging information locally.

At the implementation level it would be a great advantage to have the possibility to build each agent's logic as a separate, self-contained building block, while retaining the possibility to easily test the protection system as a whole. Moreover, it is required to use standard communication protocol, such as IEC 61850, for the communication between agents. A methodology to implement such a distributed system is now discussed.

USE OF IEC 61850 AND IEC 61499

Since IEC 61499 is a standard explicitly aimed at the development of distributed automation logics, its benefits can be readily exploited in this application: however, since another desirable feature is to support IEC 61850 for agents communication, an integration methodology is needed: a first possibility (see [1] for example) is to directly model IEC 61850 logical nodes using IEC 61499 Function Blocks, and than to develop a full set of communication Service Interface Function Blocks to implement the actual IEC 61850 communication network. However, a more 'lightweight' approach is possible if one is only interested in the subset of IEC 61850 represented by GOOSE events messaging, since in this case one can simply exploit the presence of the 'event' concept in both standards: IEC 61499 events can be directly mapped to IEC 61850 events, without the need to model the application logic around the Logical Node concept. The encapsulation capabilities of IEC 61499 can than be used to hide this mapping inside a Composite Function Block, obtaining an high level representation of the agent, in which the application logic is clearly separated from the communication part and which is easily reusable.

Figure 4 shows the basic principle of event translation between the two standards, with reference to the transmission of three distinct events.

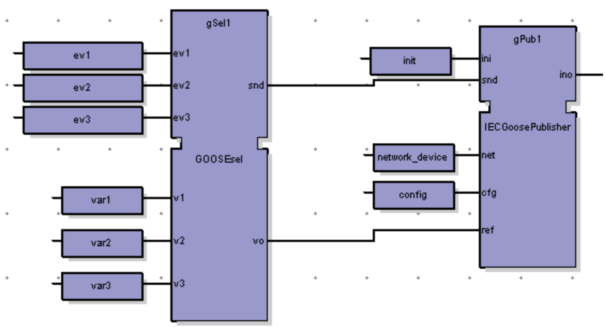

Figure 4 - Event translation

In the figure, three 61499 events (ev1, ev2, ev3) are mapped to three distinct GOOSE messages, identified through their GoCBRef (the variables var1, var2, var3 in figure). A simple multiplexer-like Function Block is used, whose output is directly connected to an IEC 61850 Service Interface Function Block, implementing the actual transmission of data.

Encapsulation

A key advantage of IEC 61499 is the possibility to build modular applications, facilitating code reuse and maintainability [4]. In particular, the separation of the agent logic from the communication part is easily realized by means of Composite Function Blocks, allowing the testing of the whole logic as a PIM (Platform Independent Model), prior to adding the IEC 61850 part and mapping to specific IEDs (PSM, Platform Specific Model).

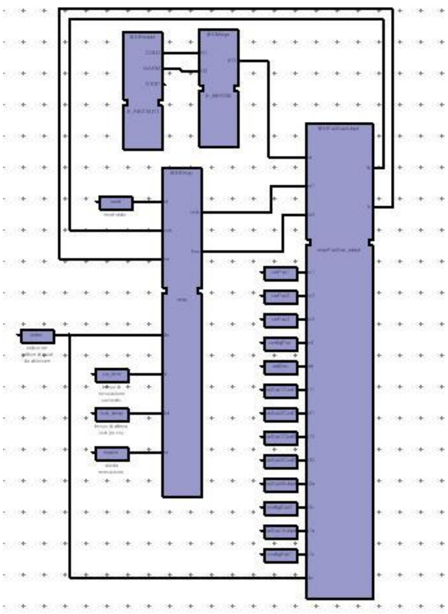

Figure 5 - Secondary substation node

Figure 5 shows the Composite Function Block implementation of a secondary substation node, with the rightmost block encapsulating IEC 61850 communication and the others blocks implementing application logic. The two parts are clearly separated and a modification of the network protocol doesn't require any reworking of the logic.

These high level blocks may then be further encapsulated into a single, top-level Composite Function Block which may be easily interconnected with other identical blocks to build the complete application. The deployment of the logic to any possible variety of network topology is thus only a matter of picking the required number of high-level blocks and interconnecting them as needed. Any modification of the agent's logic can be automatically propagated to all agents.

ACKNOWLEDGMENTS

This work has been partly funded by the Research Fund for the Italian Electrical System under the Contract Agreement between RSE (formerly known as ERSE) and the Ministry of Economic Development – General Directorate for Nuclear Energy, Renewable Energy and Energy Efficiency stipulated on July 29, 2009 in compliance with the Decree of March 19, 2009.

CONCLUSIONS

The development of auto configurable protection logics is an important topic in a context where network topology may be changing quite frequently (not only in case of faults, but also following network optimization algorithms), and where the presence of distributed generation may require the use of anti-islanding protection.

The use of agent-based schemes, along with development methodologies supporting encapsulation and model-driven architectures greatly simplifies testing and integration of such logics, facilitating maintenance, reuse of code, and extensibility. The integration between IEC 61499 and IEC 61850 can be greatly simplified by simply translating the 'event' concept, present in both standards.

REFERENCES

[1] V. Vyatkin, G. Zhabelova, 2011, "Multi-agent Smart Grid Automation Architecture based on IEC 61850/61499 Intelligent Logical Nodes", *IEEE Transactions on Industrial Electronics.*

[2] N. Higgins, V. Vyatkin, N. Nair and K. Schwarz, 2010, "Distributed Power System Automation With IEC 61850, IEC 61499, and Intelligent Control", *IEEE Transactions on Systems, Man and Cybernetics, Part C, 40(3).*

[3] V. Vyatkin, G. Zhabelova, N. Higgins, M. Ulieru, K. Schwarz, N.-K.C. Nair, 2010, "Standards-enabled Smart Grid for the future Energy Web", *Innovative Smart Grid Technologies (ISGT).*

[4] A. Zoitl, V. Vyatkin, 2009, "IEC 61499 Architecture for Distributed Automation: the 'Glass Half Full' View", *IEEE IndustrialElectronics Magazine*, 3(4), pp. 7-23.

[5] A. Cerretti, G. Scrosati, L. Consiglio, 2011, "Upgrade of Enel MV network automation to improve performances in presence of faults and to deal DG", *CIRED 2011, Frankfurt.*

[6] R. Calone, A. Cerretti, A. Fatica, 2011, "Evolution of the Fault Locator on MV distribution networks: from simple stand alone device, to a sophisticated strategic component of the SMART GRID control system", *CIRED 2011, Frankfurt.*

[7] IEC, 2005, "IEC 61499-1 Function Blocks for industrial process measurement and control systems - Architecture", *Geneva.*

[8] IEC, 2003, "IEC 61850 Communication networks and systems in substations", *Geneva.*