Design and Implementation of a Substation Automation Unit

Andrea Angioni, Anna Kulmala, Davide Della Giustina, Markus Mirz, Antti Mutanen, Alessio Dedè, Ferdinanda Ponci, Lu Shengye, Giovanni Massa, Sami Repo, and Antonello Monti

***Abstract*—The increment of distributed energy resources such as small photovoltaic plants or microcogeneration systems, together with the increase of electric vehicles connected to the grid, is bringing new challenges to distribution system operators (DSOs). In order to support DSOs to face such new challenges, this paper proposes a new solution based on a framework for the implementation of substation automation units. This framework supports the development of smart distribution systems solutions. The monitoring and control functionalities of the proposed solution are decentralized and, therefore, closer to the associated field devices compared to traditional solutions. Thus, the proposed approach allows us to increase performance and to reduce the system complexity by avoiding the concentration of data in one single point. An automation unit placed in the electrical substation is proposed and characterized for automation in distribution networks. Test results in laboratory and real-world environments will show some results of the proposed solution.**

***Index Terms*—Distribution grids, distributed monitoring and control, smart grid, substation automation.**

I. Introduction

NOWADAYS, the distribution network is hosting a growing portion of renewable generation and loads with new power profiles such as electric vehicles or heat pumps. This situation yields more frequent and higher disturbances, requiring new approaches to guarantee an adequate quality of power supply [1]. Thus, there is a growing need for automation systems characterized by advanced decentralized monitoring and control capabilities within the network [2], [3]. In order to realize such a decentralized control approach, distributed energy resources (DERs), reactive power compensators, controllable loads and on-load-tap-changer transformers (OLTCs) are participating in such automation functionalities within the network. Nevertheless, the various grid resource actors should be coordinated in order to obtain a larger observability and to control the power flow in the grid in a more efficient way. Such coordination may take place at the control center level, the so-called distribution management system (DMS), being characterized by a centralized architecture (CA) configuration [2], [4]. However, the CA may present some issues related to the time consuming execution of state estimation and control algorithms when receiving and elaborating measurements from several thousands of devices from the medium voltage (MV) and low voltage (LV) grid, as well as to send a large number of control set points. On the other hand, multi-agent based systems may share the intelligence between several units [5]; or may allow to solve power congestions and voltage issues at a decentralized microgrid level [6]. However, fully distributed automation solutions may lack in effective coordination among the different actors; particularly, when considering large distribution grids with different voltage levels (MV and LV buses with tens of thousands of nodes) [4].

Due to previous considerations, this paper proposes an automation architecture capable of allocating the intelligence at each voltage level [7], reducing the DMS computational and communication burden, while maintaining the possibility to coordinate the automation in a hierarchical way. In particular, the paper presents the implementation of substation automation units (SAUs) for primary substations (PS) and secondary substations (SS). Such a solution allows to decentralize advanced functionalities, reaching the goal of monitoring and controlling every portion of the grid supplied by a substation by means of local intelligence. Thus, PS automation units (PSAUs) are responsible for the MV network and SS automation units (SSAUs) address the need related to LV network monitoring and control. A proper coordination among the SAUs and with the DMS guarantees effective monitoring and control functionalities within shorter time frames than CA approaches and with higher robustness than the fully distributed architectures. In Fig. 1, two automation architectures, respectively the CA approach and SAU based architecture (SA) are shown. It can be seen that, for the SA approach, the DMS interacts only with the PSAU that provides a filtered quantity of information on which it bases control and supervisory actions. In this paper the SAU actor is presented and its feasibility in distribution automation is demonstrated.

Manuscript received May 3, 2016; revised July 19, 2016; accepted September 10, 2016. Date of publication September 29, 2016; date of current version March 22, 2017. This work was supported by the European Union Seventh Framework Program FP7-SMARTCITIES-2013 under grant agreement 608860 IDE4L—Ideal grid for all. Paper no. TPWRD-00602-2016.

A. Angioni, M. Mirz, F. Ponci, and A. Monti are with the E.ON Energy Research Center, ACS Institute for Automation of Complex Power Systems, RWTH University, Aachen 52074, Germany (e-mail: aangioni@eonerc.rwth-aachen.de; mmirz@eonerc.rwth-aachen.de; fponci@eonerc.rwth-aachen.de; amonti@eonerc.rwth-aachen.de).

A. Kulmala is with the VTT Technical Research Centre of Finland, Espoo 02150, Finland (e-mail: anna.kulmala@vtt.fi).

A. Mutanen, L. Shengye, and S. Repo are with the Department of Electrical Energy Engineering, Tampere University of Technology, Tampere 33101, Finland (e-mail: antti.mutanen@tut.fi; shengye.lu@tut.fi; sami.repo@vtt.fi).

D. Della Giustina, A. Dedè, and G. Massa are with the Unareti SPA, Brescia 25124, Italy. (e-mail: davide.dellagiustina@unareti.it; alessio.dede@unareti.it; giovanni.massa@unareti.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRD.2016.2614493

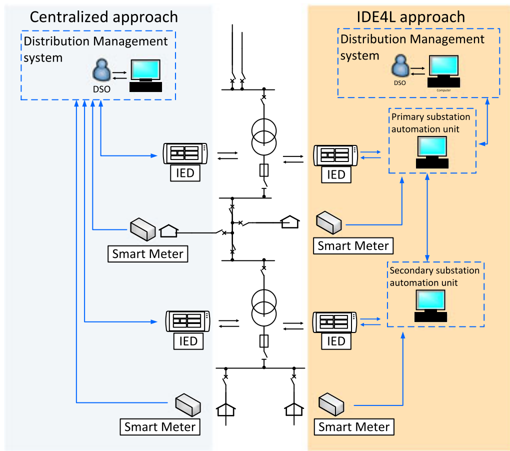

Fig. 1. Comparison between centralized and IDE4L approach.

The distribution automation functionalities are described through a comprehensive set of use cases (UCs) formatted in a standardized manner [8]. The main UCs that the SAU must perform are the collection of measurements, load and production forecasting (LF and PF), state estimation (SE), state forecasting (SF) and real-time power control (PC) [8]. Furthermore, an offline power control cost parameter update (PCPU) function and a function that coordinates the operation of cascaded transformer tap changers (block OLTCs of transformers, BOT) is implemented. The SAU implements these functionalities using a three layered structure consisting of interfaces, databases (DBs) and applications. The real-time measurements and commands are stored in a relational DB, from where the applications (e.g. state estimation, power control) access the updated data and produce the output that is again stored in the DB. Relevant data is then sent to controllable resources, other SAUs and the DMS. All data exchanges go through the DB in order to realize an effective decoupling between the communication interfaces and the algorithms. Indeed, algorithms and interfaces can be added or substituted, making the overall architecture flexible to host new software in future updates. This paper presents both the architecture concept of the SAU and an implemented instance of it. The SAU implementation covers a wide range of communication schemes that have been demonstrated to be essential to realize effective distribution automation solutions [9], [10] and is based on standard documents for data classes and communication protocols [11]–[14]. Applications (also called functions), interfaces and DBs are presented, respectively, in Sections II, III and IV, highlighting the technology neutral architecture features and the field-ready implementation details. In chapter V, the role of the SAU in the main steps of monitoring and control use cases are highlighted. Finally, in chapter VI, some lab integration tests, that show the performances of the SAU in terms of amount of data exchange, computational burden and response time, are shown together with some tests results of SE, PC (both of them in laboratory setup) and monitoring (in a real distribution grid) use cases. Such tests results are not intended to analyze the performance of each algorithm (e.g. accuracy of state estimation or solved congestions with power control) but rather to demonstrate that they may be accommodated in the SA.

II. Functions

The functions realized by the SAU are grouped in terms of monitoring and control functions. Modularity is one of the basic design principles of the SAU. Hence, all functions are implemented separately and communicate with each other only through the DB meaning that all input data is read from the DB and all output data is written to the DB. Therefore, the internal implementation of each function can be easily replaced with another implementation and the only thing that needs to remain unchanged is the DB interface.

A. Monitoring Functions

Among the monitoring functions, the SAU includes measurements collection from intelligent electronic devices (IEDs) such as smart meters (SMs), remote terminal units (RTUs) and phasor measurement units (PMUs), as well as load and production forecasting and state estimation and forecasting algorithms. The load and production forecast functions calculate power forecasts for load and generation. The load predictions are based on historic load measurements, calendar variables, such as time of day and day of week, historical temperatures and forecasted temperatures. Both, very short term and short term forecasts are performed. The very short term forecasts are made with 10-minute resolution and 30-minute forecasting horizon. The short term forecasts are made with one-hour resolution and 24–48 hour forecasting horizon. The production forecasts are calculated similarly for photovoltaic (PV) and wind generation. Wind speed, solar irradiance and temperature measurements are used for the production forecast [15]. The forecasted power values for the present time are utilized in SE as pseudo-measurements [16] if real-time power measurements are not available (e.g. SMs are not read in real-time or some customers do not have SMs at all). The forecasted power values are also exploited in SF function to determine the future network state. The state estimator determines the voltages and currents in the whole network. As inputs it uses the real-time measurement data and pseudo measurements of load and production. The SE is composed of a weighted least square (WLS) algorithm that uses branch currents as state variables [17]. The state forecast algorithm determines the network state at future time steps utilizing the outputs of the load and production forecasters and static network data. It is based on the same algorithm as the state estimator and can, therefore, handle redundant (overlapping) forecasts. It handles the uncertainties for the forecasted states in order to be used as weights in the WLS method. Depending on the input parameters from the load and production forecasters, the state forecaster calculates either very short term or short term state forecasts [17].

B. Control Functions

The control functions include the real-time power control function, the offline power control cost parameter update function and the function that coordinates the operation of cascaded OLTCs. The real-time power control aims at keeping network voltages at an acceptable level [2] and to prevent overloading of network components. It can utilize different types of resources for control purposes, such as distributed generation (DG) units, reactive power compensators, controllable loads and OLTCs. It also uses SE results, network switching state data and static network data as inputs. The PC function has been implemented as an optimization algorithm using Octave's sequential quadratic programming [18], [19]. The algorithm aims at minimizing network losses, generation curtailment cost, load control cost, number of tap changer operations and voltage variations from a set of values at each network node. It operates by the set points of primary controllers, such as automatic voltage control (AVC) relays of the OLTCs, real and reactive power DG unit controllers, reactive power controllers of reactive power compensators and real power controllers of controllable loads [19]. SE results are mandatory inputs for the PC algorithm and, therefore, the algorithms need to be executed sequentially. DB flags are used to realize the sequential operation of the algorithms.

The PCPU algorithm determines the cost parameter values used in the PC algorithm optimization. It utilizes load and production forecasts as inputs and its purpose is to prevent unnecessary control actions, especially continuous tap changer operations. The BOT function aims at preventing concurrent tap changer operations at HV/LV and MV/LV transformers. It is present only at the PSAU location and operates by sending block signals to those SAUs that should wait for the upper level SAU to operate. This function allows to avoid the overlapping of corrective actions from OLTCs at different voltage levels.

III. Communication Interfaces for SAU

To communicate with field devices and with other controllers installed in the control chain, each SAU must provide a large set of communication protocols which can be used to integrate modern and legacy systems. SAU interfaces can be divided in two clusters. 1. Operation interfaces are dedicated at exchanging data and commands with other SAUs or with the DMS. 2. Field interfaces consist of a set of protocols to communicate with field devices such as primary controllers of DERs, SMs etc. The first group of interfaces includes the communication scheme to read and write among SAUs and with the DMS. It also provides to the SAUs the possibility to publish and subscribe reports with the results of forecast and state estimation algorithms. The interfaces in the second group allow the SAU to read the reports of several types of measurement devices or to withdraw measurements in case of congestions or fault events; it also allows the SAUs to write new set points to primary controllers.

The following interfaces have been implemented for the SAU: a) MMS Server and client to allow communication with substation IEDs, among SAUs and with the DMS; b) DLMS/COSEM (Device Language Message Specification/ Companion Specification for Energy Metering) client to communicate with a new generation of SMs; c) web service clients (SOAP/REST) to retrieve data from those SMs not compliant with DLMS/COSEM standard; d) Modbus masters dedicated mainly to the communication with DERs (PV inverters, cogeneration controllers, etc.) lacking a native IEC61850 interface; e) specific IEC61850-90-5/C37.118 client to exchange data between SAUs, DMS and PMUs. In the next sections the MMS and DLMS/COSEM interfaces are presented as examples of operation and field interfaces, even though MMS is used to communicate with substation IEDs as well. MMS Interfaces

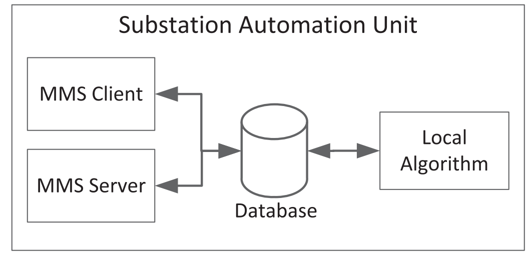

Fig. 2. MMS Interface coordination with DB and applications.

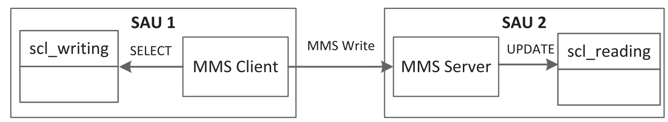

Fig. 3. Schematic of MMS write variable feature.

The MMS client and server are based on the libiec61850 library [20] and the MMSlite package [21] respectively. The coordination with the DB and applications running on the SAU is shown in Fig. 2.

MMS client and server read from or update the DB depending on the action that is locally or remotely invoked (e.g. control set point). The list of IEDs to connect to is specified together with their connection parameters (e.g. the IP, the port and authentication parameters, if any) in a dedicated table "socket" in the DB. The client connects to each IED and starts to retrieve its data model, exploring the entire hierarchy (logical device, logical node, data object, and data attributes). A general query is then performed to determine the initial status of the device. In Fig. 3, a schematic overview of the MMS write variable and reporting capability is given. The variables to be written are determined by the entries of the scl_writing table in the DB of the SAU1. After selecting all variables from this table, the client performs a writing request to the remote MMS server in SAU2. The server updates the values of the variables in its scl_reading table. After executing the write function, the MMS client sets the value of a specific column to true, which indicates the processing status of the variables. Therefore, the variables are marked as written and can be deleted from the scl_writing table and stored in the historian table on the client side.

Apart from the classical read and write communication schemes, the reports have been implemented for IED to SAU and SAU to SAU communication as well. The server reports data to the client autonomously through the report service when a trigger condition is satisfied. This condition can be event-based or periodic. This approach is more band-efficient than a more traditional polling approach (MMS read service) where the client has to ask every time to the server to send the data. The report is sent without polling by the client which, however, has to subscribe to the report. The reporting interval can be set either in the substation configuration language "cid" file or by the client once it has connected to the server. Fig. 4 presents the interaction between the components when the server sends

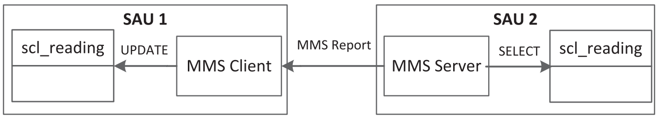

Fig. 4. Schematic of MMS reporting feature.

the reports. Before sending the dataset, the MMS server retrieves the latest values from the DB table scl_writing. Then, the data set is sent to the subscribing clients, which update their scl_reading table accordingly.

In order to make the data reading and writing process on the server side more efficient, a buffer table is added as interface between the MMS server and the scl_reading table. When the MMS server reads a value from the DB table, all the values in the table are retrieved and stored in a local set of variables. If the MMS server asks for values of other variables within a defined time interval, the values are taken from this set of variables instead of the table. This improves data related processes on the server side significantly, because less DB requests are executed and variables are read from the DB in chunks.

SM applications are now converging to the DLMS/COSEM standard [13]. The DLMS/COSEM client is based on the Gurux libraries and is implemented in Java. It is similar to the MMS client but the report service is not supported in the current version of the protocol (expected in version 2). Hence, the client has to poll the meter to get new data. Since the DB structure (described in the next chapter) is designed to be compliant with the IEC61850 data model, the client itself acts as a protocol gateway from DLMS/COSEM to IEC61850.

IV. Database for SAU

In order to perform calculations, such as state estimation, load and state forecast or power control, the SAU needs both static information, mainly related to the network topology and parameters, as well as dynamic information related to measurements and commands exchanged with field devices installed in its portion of the grid. This data is stored and managed by the SAU relational DB management system (RDBMS). The purpose of the SAU DB layer is twofold. Firstly, it provides a distributed data storage, allowing measurement data from field devices to be aggregated and processed at the substation level and to be handed over to distribution system operator (DSO) control center later. This makes the data acquisition at the MV and LV network level more effective, both in terms of higher rate and amount of information per node, without overloading the DMS. Secondly, the SAU DB layer provides a loose-coupled way to integrate the SAU interfaces that fetch measurements from field devices and store them into the DB with the control algorithms that can read commands and measurements from the field. The algorithm results, typically set points and commands, are written into a specific table of the DB (scl_writing), from where the SAU interfaces retrieve them and dispatch to relevant field devices or other SAUs. All inputs and outputs pass through the DB, which, becoming a middleware of the SAU internal architecture, avoids direct communications and data exchange between SAU components. This significantly reduces system integration costs, facilitating replacement or addition of new SAU components.

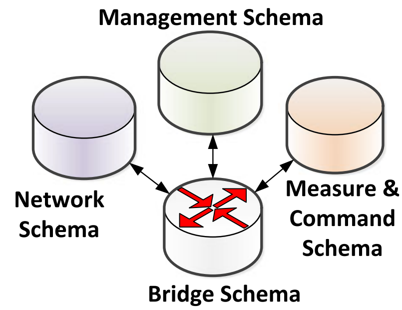

Fig. 5. Relations among SAU DB modules.

To further facilitate the integration of the whole system, most data in the database is represented according to standard data models IEC 61850 [11] and Common Information Model (CIM) [12], [14]. Data stored in the database mainly includes:

1) real-time measurements from field devices;
2) calculation results from state estimation, state forecast, and load and production forecast, etc;
3) set points and commands calculated by control functions like power control algorithm;
4) network static information about the portion of the grid controlled and monitored by the specific SAU, including network topology, asset parameters, etc;
5) management information for SAU functions, including execution logs, flags for inter-algorithm concurrency, etc.

Among this data, the first three items are mapped into the IEC 61850 data model, using standards IEC 61850-7-3, IEC 61850-7-4 and IEC 61850-7-420. The fourth item is presented as CIM model, based on standards IEC 61970-301 and IEC 61968-11. The fifth has been defined within the IDE4L project by the algorithm developers, and represents a contribution that may support future standardization committees. The database architecture implemented in the IDE4L lab experiments and field demonstrations consists of four schemas, as described in Fig. 5:

1) Measure&Command Schema – storing dynamic data including measurements, control signals, state estimation results, etc;
2) Network Schema – containing network static information such as topology and line parameters;
3) Bridge Schema – describing every relation between Measure&Command module and Network module;
4) Management Schema – managing and coordinating SAU functions.

All the data stored in Measure&Command Schema has been mapped to the IEC 61850 data model. Each measurement point or control signal is identified as a combination of physical device, logical device, logical node, data object and data attribute. For each data attribute, a set of reading values or writing values can be associated in case of measurements acquired from field IEDs or commands/set points that must be sent. Typically, SM data, substation measurements, SE and SF results are

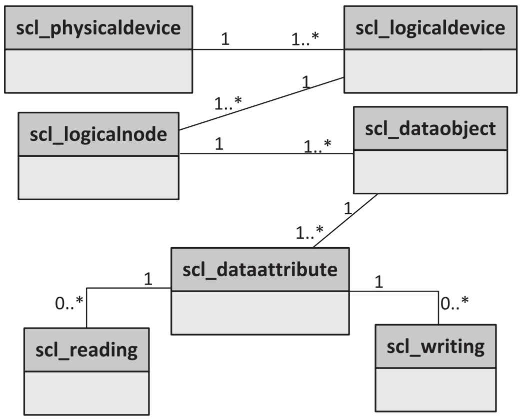

Fig. 6. DB tables for storing dynamic data about measurements and commands exchanged with field IEDs.

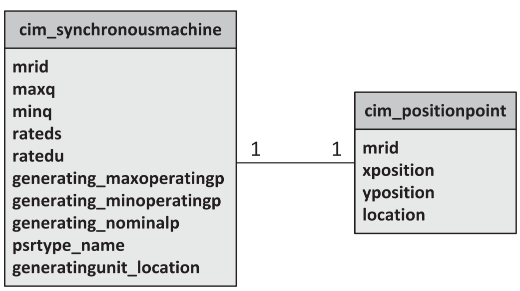

Fig. 7. DB tables to store static data concerning DGs.

modelled by logical node types MMXU, MMXN, MMTR (these are logical node names and not acronyms), etc.; control signals are described by logical node types such as DRCC and ATCC. The actual modelling depends on specific IEDs used in each demonstration site. Fig. 6 describes the section of the Measure&Command schema used to store real-time measurements and commands following the IEC61850 data model. In addition to scl_reading and scl_writing entities, other entities in relation with scl_data-attribute entity have been defined to store and manage historical and forecasted profiles for measurements and commands.

The DB Schema of network model is designed according to the IEC CIM standard. The CIM standard defines a semantic model describing the components of a power system at an electrical level and the relationships between components. In the Network Schema, each table is based on one or more CIM classes, with each column corresponding to an attribute in the CIM class. One table might refer to another table, which corresponds to associations among CIM classes. The table design tries to balance the compliance of the CIM model and the simplicity of its implementation. As an example, Fig. 7 depicts the design of two tables. The table "cim_synchronousmachine" is used to store static information about every DG, e.g., maximum and minimum reactive power limit, apparent power rating, etc. This table is based on the CIM classes SynchronousMachine, GeneratingUnit and Location. The other table, "cim_positionpoint", stores the geographic coordinator information for each DG. It is based on the CIM class PositionPoint. There is a reference relationship between these two tables.

The purpose of the Bridge Schema is to connect real-time and historical measurements and commands – defined by the IEC61850 data model and stored in Measure&Command Schema – with their locations in the Network topology, which is described by the CIM model. Considering the middleware functionalities handled by the DB, a Management Schema has been designed in order to coordinate and control the execution of the SAU algorithms. The Management Schema can store also logs and information which can be used for debugging purposes or to evaluate system performance. The SAU DB is hosted on the same computer as other SAU components (i.e., SAU interfaces and functions). It is implemented using object-RDBMS implemented by means of PostgreSQL [22].

V. Example Use Cases: SE and PC

SE and PC functions are executed sequentially either at fixed time intervals or in case of congestions or fault events. Their sequential execution is mainly due to the fact that SE results are needed as an input by the PC algorithm. At the beginning of the execution interval, SE reads its input data from the DB. Real-time measurements are written to the DB by the interfaces implementing the monitoring function, while LF and PF provide the pseudo-measurements. MV network SE, realized at the PSAU level, utilizes also the MV/LV transformer power values calculated by the LV network SE. This data is transferred from the SSAUs to the PSAU using the SAU-SAU MMS interface. The switching state data is available at the SAU DB and updated regularly by the DMS. The network model, used by SE and PC, is updated only if the network data or the switching states change. After reading the input data, the SE performs its internal calculations to determine the network state. The results are written to the SAU DB and also the relevant DB flags are updated, so that other algorithms know when the SE is ready. When the flag indicating that the SE is ready is raised, the PC algorithm reads the SE results from the DB and starts its internal calculations to determine the optimal set points for primary controllers. The new set points are written to the DB if the enhancement of the network state is considered adequate compared to the previous set points. In the previous case, the interfaces between the DB and the IEDs send the new set points to the IEDs. If the PC algorithm is not able to find set points that keep the network state acceptable, a help request is sent to the upper level SAU or to the DMS. A SSAU PC help request is sent to PSAU using the MMS SAU-SAU interface. The MV network PC algorithm reads the possible help requests from the PSAU DB and utilizes them when determining the MV network resource set points. The PSAU PC algorithm help request is sent to the control center, invoking the network reconfiguration algorithm [19].

VI. Tests

The SAU proposed in this paper has been evaluated through:

1) a set of lab integration tests;

TABLE I
REGULAR INFORMATION EXCHANGE FOR CENTRALIZED AND IDE4L APPROACH

UC	IPCA	IRCA	IPSA	IRSA	Information exchange	Amount of data	Reporting rate (frame/s)	Traffic CA DMS (B/s)	Traffic SA DMS (B/s)	Traffic SA PSAU (B/s)	Traffic SA SSAU (B/s)
LV Mon.	IED, SM	DMS	IED	SSAU	3ph V, P, Q measurements and connection status	12 for each node	0.02	60000	–	–	60
LV Mon.	–	–	DMS	PSAU	SWI-BRE status	3 for each node	0.0001	–	18.75	18.75	0.075
LV Mon.	–	–	PSAU	SSAU	SWI-BRE status	3 for each node	0.0001	–	–	18.75	0.075
LV Mon.	–	–	SSAU	PSAU	SWI-BRE status	3 for each node	0.001	–	–	187.5	0.75
LV Mon.	–	–	PSAU	DMS	SWI-BRE status	3 for each node	0.001	–	187.5	187.5	0.75
MV Mon.	IED, SM	DMS	IED	PSAU	3ph V, P, Q measurements and connection status	12 for each node	0.02	60	–	60	–
MV Mon.	–	–	DMS	PSAU	SWI-BRE status	3 for each node	0.0001	–	0.075	0.075	–
MV Mon.	–	–	PSAU	DMS	SWI-BRE status	3 for each node	0.001	–	0.075	0.075	–
LV SE	–	–	SSAU	PSAU	Estimation at point of connection (V, P, Q)	9	0.02	–	–	45	0.18
MV SE	–	–	PSAU	SSAU	Estimation at point of connection (V, P, Q)	9	0.02	–	–	45	0.18
MV SE	–	–	PSAU	DMS	Result estimation (V, P, Q) + RMS current	9 for each node	0.02	–	45	45	–
LV forecst	–	–	SSAU	PSAU	Forecast point of connection for 24 hours	216	0.001	–	–	54	0.216
MV forecst	–	–	PSAU	SSAU	Forecast point of connection for 24 hours	216	0.001	–	–	54	0.216
MV forecst	–	–	PSAU	DMS	result forecast (V, P, Q) for 24 hours	216 for each node	0.001	–	54	54	–

2) testing of SE, PC and Monitoring use cases.

The lab integration tests show the characteristics of data exchanges (VI-A), computational burden (VI-B), disk space usage of SAU computer (VI-C) and the response time to congestion events (VI-D). The lab integration tests are meant to evaluate the coordination among the three layers of the SAU, as well as among SAUs and between SAUs and IEDs. Some of the requirements are compared with the ones of a fully CA in order to evaluate the advantages of distributing the intelligence at substation levels. This permits to draw a first characterization of the SA and the SAU actor. The testing of SE (VI-E) and PC (VI-F) use cases, in laboratory environment, and monitoring use case (VI-G) in an Italian Multi-Utility Company, Unareti SpA, which operates in the areas of Milan and Brescia, Northern Italy, is intended to prove that advanced automation functionalities, may be obtained through the SA.

A. Requirements for Data Exchange

The first analysis is based on the information exchange required to execute the monitoring use case (measurement collection, SE, LF, PF and SF). The information exchange is referred to the Unareti demonstration site and the results are presented in Table I. The test grid consists of 3 MV feeders connected to a HV/MV transformer in the PS. The MV grid has a total of 250 MV buses, each one hosting a MV/LV transformer with a LV grid with 250 buses. Only one LV grid is used for demonstration; however, for the sake of simplicity, it is assumed that each MV/LV transformer is connected to a LV grid with 250 LV buses. The second and third columns represent respectively the information producer and receiver for the case of the CA (IPCA and IRCA), whereas the third and the fourth ones represent the information producer and receiver for the case of the SA (IPSA and IRSA). Each row includes the description of the information exchange that is required for a use case, the amount of data included and the generated traffic considering the required reporting rate. In the first row it can be seen that the 3 phase

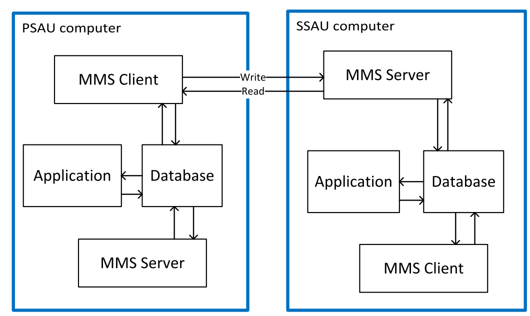

Fig. 8. RWTH laboratory test setup.

RMS voltage, the active and reactive power and the connection status are reported regularly to the SSAU, generating a reporting rate of one frame every 50 seconds. A traffic of 60 B/s is measured for the SSAU case, as shown in column "traffic SA SSAU (B/s)", while a traffic of 60 kB/s is detected for the CA case. It has to be considered that such CA traffic has to be scaled for larger distribution system, as the portion tested in the IDE4L project is only a small fraction of the Unareti distribution grid. Instead, for the SA approach, the amount of the traffic that each SAU has to deal with is only related to the size of its own MV and LV grid. The total traffic to be managed by the DMS for the CA approach is 60060 B/s, whereas for the SA approach it is 305.4 B/s for the DMS, 769.65 B/s for the PSAU and 62.44 B/s for the SSAU. Similar results are obtained in case of events such as faults or congestions when the PC or the fault location algorithms start a significant exchange of information, subject to strict delay requirements (for instance in IEC61850-5).

The communication among SAUs has been also tested in the lab environment shown in Fig. 8 [23]. The writing and reading communication schemes through MMS protocol for 400 measurement and control units, from the PSAU client to the SSAU server, have been tested. The data exchange shows proper behavior for reading and writing rates in the order of seconds.

TABLE II
EXECUTION TIME FOR SE USE CASE WITH SAU COMPUTER AT TAMPERE UNIVERSITY OF TECHNOLOGY (TUT) LABORATORY

Operation	Time (s)
Algorithm initialization (done only once)	0.140
Algorithm reset (only if changes in static network data)	
• Read static network data	0.006
Algorithm execution	
• Check if the static network data has been updated	0.001
• Read breaker statuses and compare to previous statuses	0.002
• Update network configuration (only if changes in breaker statuses)	0.069
• Read load and production pseudo-measurements	0.012
• Read real-time measurements (IEDs and SMs)	0.062
• Combine pseudo- and real-time measurements	0.009
• Calculate the state estimates	0.277
• Model the distribution transformer	0.007
• Write the results to the database	0.530
• Update flags and write log messages	0.009

In particular, the SAU computers (both equipped with Windows 7 PC, i5-2400 CPU and 4 GB RAM) spend always less than 50 and 300 milliseconds respectively to read and write a block of 400 data from/to the DB

B. Requirements on Computational Burden

The SAU continuously performs measurement collection, writing and reading from the DB and execution of algorithms, such as SE and PC. In the following test, the average execution times of individual operations included in the SE are evaluated. A small portion of the Unareti distribution grid consisting of 15 LV buses has been periodically estimated. Table II shows the execution time for each operation performed on the SAU computer, that in this test case has been composed by a Windows 7 PC with Intel i7-2600 processor, 16 GB RAM and 512 GB SSD.

It can be seen that the operations requiring the higher time intervals are the storing of results in DB (0.530 s) and the actual calculation of the SEs (0.277 s). Due to the limitations of Octave database package, writing results to the DB is very slow. The calculation of SEs, as well as the SF, requires the matrix coefficient inversions, which is computationally heavy, being its size proportional to the size of the network. It is possible to exploit the sparsity of the coefficient matrix to speed up the inversion (the order of sparsity depends on the network topology). Nevertheless, in general, larger networks require longer execution time. On the same computer, the total execution time for the 271 bus LV network used in Unareti field demonstration is 6.3 seconds, out of which 4.1 seconds is spent writing the results to the DB and 1.3 seconds calculating the SEs.

C. Database Disk Space Usage

The SE, measurement reading and PC use cases are tested on a real time simulated portion of the Unareti distribution grid, consisting of 15 LV buses in the real time lab setup presented in Fig. 9.

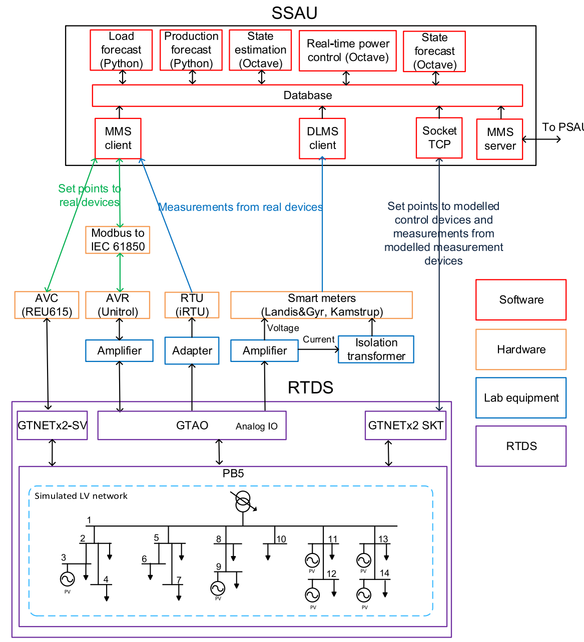

Fig. 9. The simulation arrangement at TUT RTDS laboratory.

TABLE III
DISK SPACE USAGE IN SSAU DURING THE RTDS SIMULATIONS

Function	Disk space usage
Monitoring	1.9 MB/hour
State estimation (with history)	15.6 MB/hour
Power control (with history)	0.3 MB/hour

Some real measurement devices are connected through real time digital simulator (RTDS) interfaces and provide measurements from RTDS. The measurement configuration includes power, current and voltage measurements at the LV side of the SS, obtained from a real RTU, two real SMs (Landis + Gyr and Kamstrup) measuring the power values at the load nodes, and six virtual generation node power and voltage measurements, directly read from the RTDS through the socket interface. The substation measurements are characterized by 5 second reading frequency, while the SMs and virtual measurements have a 60 second reading frequency. Controllable resources include the OLTC of the MV/LV transformer and real and reactive powers from PV units. The OLTC is controlled by an ABB REU615 AVC relay and the PV controllers are modeled in the RTDS. During the execution of the automation functions, all the measurements, control commands and algorithm outputs are stored to the SAU DB. Table III shows how much data is stored to the DB by different functions. The SE is executed when a new RTU measurement arrives at the DB, whereas PC is executed once a minute or when network congestion is detected.

The majority of the data storage is required by SE and PC algorithms to archive historical results. Historians are needed to feed forecast algorithm and posteriori statistical analysis of the network behavior. Without considering the historian tables, the

TABLE IV
RESPONSE TIME TO NETWORK CONGESTION EVENT

Steps	Time (s)
RTU sends measurements to MMS client	0–5∗
MMS client receives RTU measurements and stores them to SSAU database	< 1.0
SE detects the new measurements, calculates new state estimates, detects congestion and sets PC execution request flag to true	0.9
PC notices execution request flag and calculates new set points for generator maximum outputs	8.6
Socket TCP sends new set points to generator controllers	0–5∗
Total response time (average)	**15.5**

∗ RTU and socket TCP operate on a 5 second refresh rate.

SE and PC algorithms consume a small portion of the total storage space. Moreover, the majority of the disk space usage is ascribable by monitoring functions, as MMS and DLMS/COSEM clients. The disk space needed to store the measurements in SSAU is about 17 GB (it includes historical data of measurements and control set points) per year. While the data volume stored in the SSAU database is large, the amount of data sent to the PSAU is smaller, only about 2.7 GB per year, and only a fraction of that needs to be stored permanently. The volume of data stored into the SSAU DB grows as the number of monitored network nodes increases, but the amount of data transferred to PSAU is constant (it includes only the data of the connection point between MV and LV).

D. Response Time to Network Congestion

The SAU functions can detect and solve network congestions causing current or voltage limit violations. Table IV shows the response time, due to an event where excessive generation on “feeder 3” has caused a thermal overload on a line section of the real-time simulated 15 LV buses Unareti grid. The laboratory infrastructure exploited for this test is the one depicted in Fig. 9. The SE enables the detection of the congestion and the PC is able to solve it by limiting the generators output. The SSAU manages to solve the congestion in an average time of 15.5 s and with minimal production curtailment. Although fast, the above achieved response time is dependent on the update rate of the SE, which in this case was the same as the reporting rate (5 seconds) of the fastest measurement devices.

E. Time Domain Operation of SE

The time domain operation of the SE is presented. The simulation arrangement presented in Fig. 9 was used.

At each bus, loads are kept constant and equal to a reference average power (obtained averaging a set of actual historical data from Unareti DB, representative of typical LV lload) and productions are varied multiplying an average power (also obtained as a representative average generated power from Unareti DB) by a factor defined in Table V. Fig. 10 shows how the estimated and true (from simulation). The steady state estimates are accurate but after each change in generator output, there is some delay before the SE detects the new changed network state. In this simulation the SE was configured to be executed only once

TABLE V
SIMULATION SEQUENCE

Time	DG output factor
0 s	1.0
150 s	0.3
270 s	1
510 s	0.3

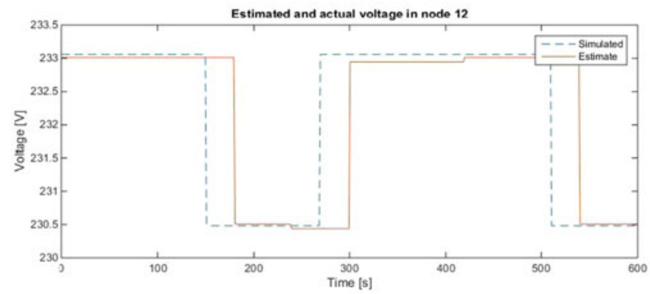

Fig. 10. Example of time domain operation of the SE algorithm.

TABLE VI
SIMULATION SEQUENCE

Time	DG output factor	Loading conditions
0 s	3	MIN
50 s	0.9	MIN
230 s	0.9	MAX
410 s	3	MAX

a minute, which corresponds to the SE execution interval used in the field Unareti demonstration.

F. Time Domain Operation of PC

The response times of different SAU algorithms in case of feeder overloading are presented in chapter VI-D. This Section presents the time domain operation of the algorithms in case of voltage limit violation, using the laboratory infrastructure and simulation network of Fig. 9. The simulation considers, in each bus, loading and generation conditions in the network as presented in Table VI. The minimum and maximum loading conditions are determined based on real low voltage SM data from Unareti DB. Generation values are obtained by multiplying an average generated power (obtained from analysis of SM data in Unareti DB) by a factor defined in Table VI. The voltage limits have been set to $\pm$ 5%. The PC algorithm objective function has been formulated to minimize network losses and generation curtailment. The controllable resources are the MV/LV transformer OLTC and the real and reactive powers of the 6 PV units (as depicted in Fig. 9). Both the SE and PC are executed once a minute.

The time domain simulation results are shown in Fig. 11. The simulation results show that the PC algorithm is able to restore network voltages to an acceptable level by controlling the OLTC and reactive powers of PV units. Active power

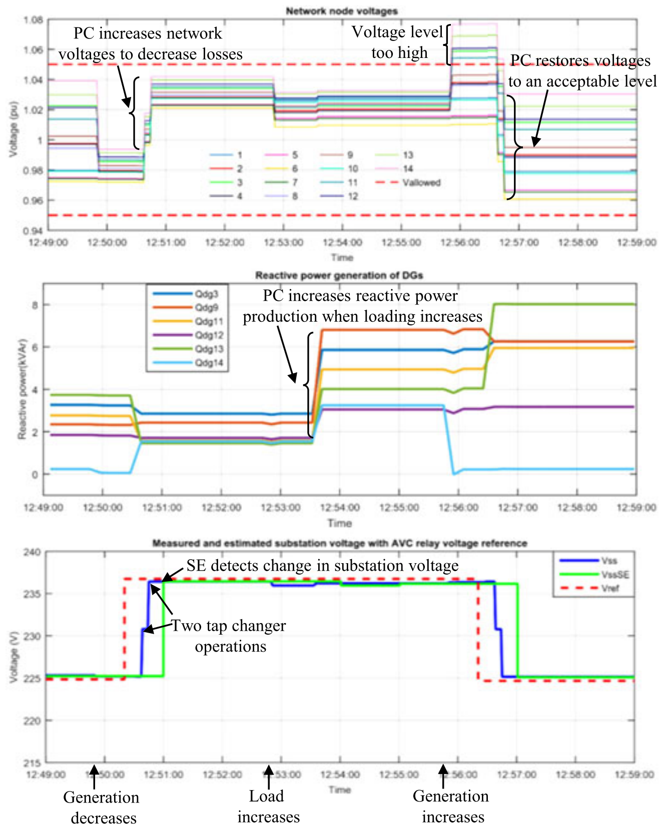

Fig. 11. Time domain operation of the PC algorithm. The uppermost figure presents network voltages, the second figure shows reactive power production of the controllable units and the third figure depicts the measured and estimated substation voltage and the AVC relay voltage reference.

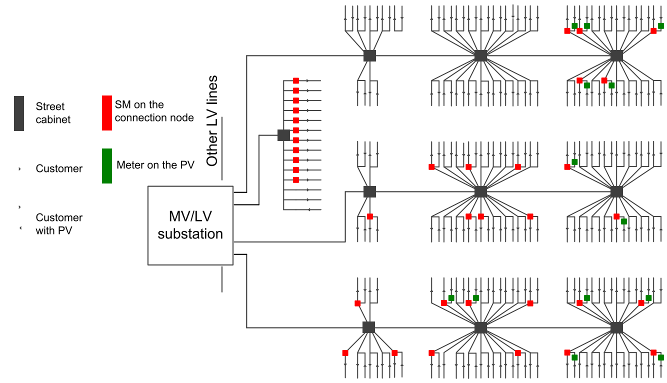

Fig. 12. A simplified single-line diagram of the Unareti's LV field test.

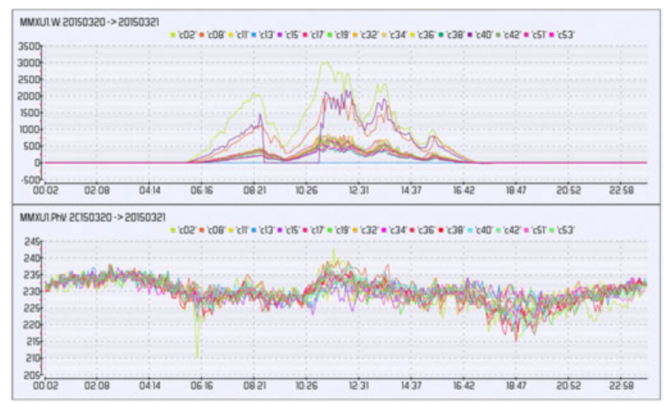

Fig. 13. Active power injected by some residential PV panels (top) and phase voltage (bottom) at the connection node.

generation curtailment was not needed. The losses are minimized by increasing the network voltage level (but still within acceptable security ranges) and by producing part of the reactive power consumed by the loads with PV units (thus decreasing reactive power flow in the LV lines).

G. Field Demonstration

The SAU based architecture is being tested also in the real Unareti distribution network. Concerning the MV level field test, it involves a PS equipped with a PSAU and three MV feeders (2 fully automated ones and the third one included only at the simulation stage). The PSAU is interfaced to the automated feeders to test four use-cases: real-time monitoring, SE, network reconfiguration, and fault location, isolation and supply restoration (FLISR). The Unareti LV level field test concerns a MV/LV substation equipped with a SSAU and connected to ten LV feeders, as depicted in Fig. 12. The SSAU is interfaced to the DSO control center through the PSAU and implements the algorithms necessary to test the following use-cases: real-time monitoring, LF and PF, SE and PC. The SSAU receives data from SMs without passing through a SM concentrator in DLMS/COSEM format and maps the customers' data to IEC61850, according to the procedure defined within Section III. Furthermore, it maps the current sensors connected to the MV and LV sides of the MV/LV transformer.

As an example, Fig. 12 reports the active power injected by some residential PV panels (top) and phase voltage (bottom) on the connection node. On March 20th, 2015, there was an eclipse which caused an abrupt reduction of the power injected into the grid, despite the fact it was a partially cloudy day in that area, as it can be noted in the figure. The power reduction was in the range of 40%-75%. It can be seen that, related to this power reduction, there was also a voltage reduction of about 5 V in the worst-case. The data shown in Fig. 13 have been collected by the DMS from the SSAU, exploiting the SAU based architecture. The detection and evaluation of such an event is enabled by the new proposed architecture.

VII. CONCLUSION

The SAU role in the distribution grid and its architecture has been presented and some instances have been implemented and tested with regards to the types of data exchange required and the traffic generated. The computational burden required during the automation and the response time in case of a congestion events have been evaluated. Tests results shown in the paper represent a preliminary demonstration of the feasibility of the SAU based architecture. It is demonstrated that, due to the small size of network to be automatized by each SAU (thanks to the splitting of the automation in smaller areas related to the MV and LV grids) and the effective communication schemes and database design, the SA behaves properly in the monitoring and control functions described in the distribution automation use cases. In particular, the data exchanges requirements are relaxed for SA compared to centralized architecture, and computational burden and disk space usage also show to be affordable, also for low cost computation units. SE and PC use cases, show to be properly integrated in the SAU based architecture and to perform as expected. The SA was presented as technology neutral architecture and only after, demonstrated in several technology instances, both in lab and field. Further instances of hardware and software, both for SAUs and IEDs may be tested in order to evaluate their impact in terms of performance of SE and PC. Anyway, the very promising results obtained by the developed concept and implementation allows to demonstrate the effectiveness and robustness of the proposed solution for decentralized control based smart grid applications.

References

[1] R. A. Walling, R. Saint, R. C. Dugan, J. Burke, and L. A. Kojovic, "Summary of distributed resources impact on power delivery systems," *IEEE Trans. Power Del.*, vol. 23, no. 3, pp. 1636–1644, Jul. 2008.

[2] J. Northcote-Green and R. Williams, *Control and Automation of Electrical Power Distribution System*, vol. I, Boca Raton, FL, USA: Taylor & Francis, 2007.

[3] "Siemens ENEAS solutions for distribution automation," 2016. [Online]. Available: www.siemens.com/download?DLA01_388

[4] I. Roytelman and V. Ganesan, "Coordinated local and centralized control in distribution management systems," *IEEE Trans. Power Del.*, vol. 15, no. 2, pp. 718–724, Apr. 2000.

[5] I.-H. Lim *et al.*, "Design and implementation of multiagent-based distributed restoration system in DAS," *IEEE Trans. Power Del.*, vol. 28, no. 2, pp. 585–593, Apr. 2013.

[6] T. L. Vandoorn, J. D. M. De Kooning, B. Meersman, and L. Vandevelde, "Voltage-based droop control of renewables to avoid on–off oscillations caused by overvoltages," *IEEE Trans. Power Del.*, vol. 28, no. 2, pp. 845–854, Apr. 2013.

[7] S. Lu, S. Repo, and D. Della Giustina, "Standard-based secondary substation automation unit—The ICT perspective," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Eur.*, Oct. 2014, pp. 1–6.

[8] IDE4L, deliverable 3.2, "Architecture design and implementation," 2015. [Online]. Available: http://ide4l.eu/results/

[9] K. Myongsoo and J. J. Metzner, "A key exchange method for intelligent electronic devices in distribution automation," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1458–1464, Jul. 2010.

[10] W. Ling, D. Liu, Y. Lu, P. Du, and F. Pan, "IEC 61850 model expansion toward distributed fault localization, isolation, and supply restoration," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 977–984, Jun. 2014.

[11] *Communication Networks and Systems for Power Utility Automation*, IEC Standard 61850, 2015.

[12] *Energy Management System Application Program Interface*, IEC Standard 61970, 2015.

[13] *Electricity Metering—Data Exchange for Meter Reading, Tariff and Load Control*, IEC Standard 62056, 2015.

[14] *Common Information Model (CIM)/Distribution Management*, IEC Standard 61968, 2008.

[15] IDE4L, deliverable 5.1, "State estimation and forecasting algorithms on MV & LV networks," 2015. [Online]. Available: http://ide4l.eu/results/

[16] A. Angioni, T. Schlosser, F. Ponci, and A. Monti, "Impact of pseudo-measurements from new power profiles on state estimation in low-voltage grids," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 1, pp. 70–77, Jan. 2016.

[17] A. Mutanen, A. Koto, A. Kulmala, and P. Järventausta, "Development and testing of a branch current based distribution system state estimator," in *Proc. 46th Int. Univ. Power Eng. Conf.*, Soest, Germany, Sep. 5–8, 2011, pp. 1–6.

[18] A. Kulmala, S. Repo, and P. Järventausta, "Coordinated voltage control in distribution networks including several distributed energy resources," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 2010–2020, Jul. 2014.

[19] IDE4L, deliverable 5.2/3 "Congestion management in distribution networks," 2015. [Online]. Available: http://ide4l.eu/results/

[20] libiec61850, 2016. [Online]. Available: http://libiec61850.com/libiec61850/

[21] MMSlite package, 2016. [Online]. Available: http://www.ama-group.de/home.html

[22] PostgreSQL, 2016. [Online]. Available: http://www.postgresql.org/docs/9.2/static/app-psql.html

[23] A. Sadu, A. Angioni, J. Liu, F. Ponci, and A. Monti, "A platform for testing monitoring systems for the power distribution grid," in *Proc. IEEE Int. Workshop Appl. Meas. Power Syst. Proc.*, Aachen, Germany, 2014, pp. 1–6.

Andrea Angioni, photograph and biography not available at the time of publication.

Anna Kulmala, photograph and biography not available at the time of publication.

Davide Della Giustina, photograph and biography not available at the time of publication.

Markus Mirz, photograph and biography not available at the time of publication.

Antti Mutanen, photograph and biography not available at the time of publication.

Alessio Dedé, photograph and biography not available at the time of publication.

Ferdinanda Ponci, photograph and biography not available at the time of publication.

Lu Shengye, photograph and biography not available at the time of publication.

Giovanni Massa, photograph and biography not available at the time of publication.

Sami Repo, photograph and biography not available at the time of publication.

Antonello Monti, photograph and biography not available at the time of publication.