INTERFACING WITH SCADA SYSTEM FOR ENERGY MANAGEMENT IN MULTIPLE ENERGY CARRIER INFRASTRUCTURES

Nikola Tomašević, Marko Batić, Sanja Vraneš
University of Belgrade, Institute Mihajlo Pupin

Abstract – *In order to provide more advanced and intelligent energy management systems (EMS), generic and scalable solutions for interfacing with numerous proprietary monitoring equipment and legacy SCADA systems are needed. Such solutions should provide for an easy, plug-and-play coupling with the existing monitoring devices of target infrastructure, thus supporting the integration and interoperability of the overall system. One way of providing such holistic interface that will enable energy data provision to the EMS for further evaluation is presented in this paper. The proposed solution was based on the OpenMUC framework implementing a web-service based technology for data delivery. For testing the system prior to an actual deployment, data simulation and scheduling component was developed to provide a low-level data/signal emulation which would replace signals from the field sensors, monitoring equipment, SCADAs, etc. Furthermore, the proposed interface was implemented in a scalable and adaptable manner so it could be easily adjusted and deployed, practically, within any complex infrastructure of interest. Finally, for demonstration and validation purposes, the campus of the Institute Mihajlo Pupin in Belgrade was taken as a test-bed platform which was equipped with SCADA View6000 system serving as the main building management system.*

1. INTRODUCTION

Contemporary energy management systems (EMS) usually have to supervise and control a number of technical systems and devices, including different energy production, storage and consumption units, within the target building or infrastructure. In such a case, interfacing with the numerous field-level devices, coming from different vendors, using different proprietary communication protocols, is a difficult problem to solve, particularly in case of complex infrastructures with multiple energy carrier supply. Therefore, in order to support more advanced and intelligent EMSs, generic and scalable solutions are needed to be provided in terms of easy, plug-and-play interfacing with different legacy field-level devices, monitoring equipment, etc. Such solutions should provide means for integration with the existing energy related devices and systems, but also to the legacy building management systems (BMS) such as, for instance, widely utilized systems for supervisory control and data acquisition (SCADA). So far, a number of results on interfacing with SCADA and underlying monitoring infrastructure to support energy management were published in the literature. For instance, in [1], interfacing between smart metering devices and SCADA system on the carrier level was investigated for private households and small enterprises. A concept design and communication interfaces with data monitoring system underlying the hybrid renewable energy systems were presented in [2] and [3]. Integration interfaces between new EMS centre and the existing SCADA system respecting the data acquisition and communication aspects were described in [4]. Then, a scalable, distributed SCADA systems for energy management of chain of hydel power stations and power distribution network buildings were proposed in [5] and [6], respectively. Authors of [7] proposed an integrated system for the monitoring and data acquisition related to the energy consumption of equipment or facility. Moreover, the requirements for SCADA systems in terms of standardized protocols for communication and interfaces for large wind farms were discussed in [8]. Finally, an overview of the open architecture for the EMS control centre modernization and consolidation providing modular interfacing with the new applications was provided in [9]. All these results are to the significant extent tailored to the specific scenarios of interest and do not provide a generic and scalable solution.

The objective of this paper was to present one way of providing the interface and the communication means between the energy data acquisition system (such as SCADA system) and EMS of the target infrastructure/building. Through the developed interface, the energy production and consumption data (such as electrical and thermal (hot water) energy), were provided to the EMS for further evaluation in order to optimize the energy flows (both production and consumption) within the target infrastructure [10],[11]. The proposed solution for interfacing with SCADA system was based on the OpenMUC framework [12] which was leveraged upon OSGi specifications and web-service based technology for data delivery. Supported by OpenMUC framework, the interfacing module provided the energy production and consumption data to the rest of EMS components (such as energy optimization module [11]) in a web-service based manner, on demand by querying SCADA database, in flexible and secured manner. Furthermore, different communication protocols and flexible time resolution of acquired data are supported. Apart from providing the acquired data to the EMS, the interface module also provides means for execution of the field device controls (for controlling the actuators, definition of target set-points, etc.). For validation and demonstration purposes of the proposed interface, the campus of the Institute Mihajlo Pupin (PUPIN) in Belgrade was taken as a test-bed platform with multiple energy carrier supply. At the PUPIN campus, SCADA View6000 system [13] is operating as the main BMS providing supervision and control of underlying energy related systems. Currently, it supervises both electricity production (by PV plant) and consumption, while monitoring of the thermal energy consumption (hot water produced by burning the fuel oil-mazut) is envisioned for the near future by integrating already installed digitalized flow meters/calorimeters. Nevertheless, contrary to the existing solutions, the proposed interface was envisioned in flexible and scalable manner so it could be easily extended to encompass subsequently defined metering points.

In order to properly validate and test the developed interface, before the actual deployment, a software component was developed responsible for the low-level (measurement) data simulation and scheduling. In other words, this data simulation and scheduling component, i.e. a data emulator provides the emulation of signals coming from the field sensors, SCADAs, etc. In this way, early prototyping and testing of the EMS and energy optimization modules for multi-carrier interconnected entities was made possible [10]. Manual definition of different data types/signals was also supported. At the same time, this component provides a flexible way for use case scenarios definition. This included the definition of low-level signals, patterns as a set of low-level signals and high-level use case scenarios definition as a set of signal patterns. Data/signals emulated by data emulator component are made accessible to all EMS components and provided in a web-service based manner. Moreover, it supports flexible time resolution/granularity of the output data.

The remainder of this paper is organized as follows. Section 2 specifies the main functionalities which should be supported by the interface module by analysing the interface rationale and managed field-level data. Then, the technological background of the interface was defined through the investigation of interface deployment diagram in Section 3. Section 4 specifies some of the main functionalities, technological background and integration design of the data emulator in terms of field-data simulation and scheduling. Integration and validation of the interface module were investigated subsequently in Section 5. Finally, some concluding remarks were presented in Section 6.

2. SPECIFICATION OF INTERFACE MODULE FUNCTIONALITIES

In order to provide the energy data for further processing, the proposed interface should enable a two-way communication between the energy data acquisition system (such as SCADA system) and EMS of the target infrastructure. This interface module was intended to provide all the information related to the energy consumption of target infrastructure to the rest of the EMS as well as to the energy optimization module [11]. Apart from the information on consumption of energy carriers used to supply the target infrastructure this considered also the acquisition of energy production data (if production units are deployed).

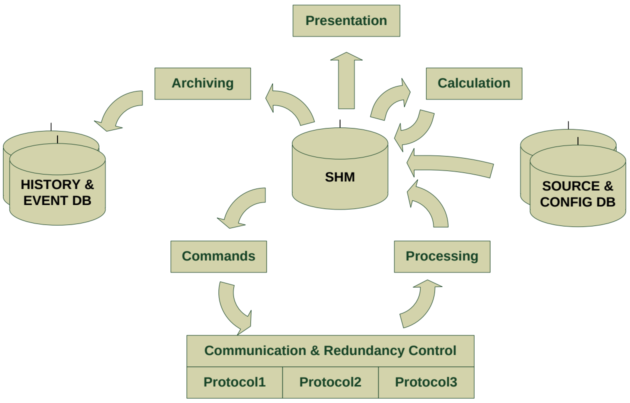

Figure 1: SCADA system architecture (View6000)

For demonstration purposes, the PUPIN campus was taken as a multiple energy carrier supply test-bed platform where SCADA View6000 system [13], shown in Figure 1, is operating as the main BMS. Currently, only electrical energy production and consumption, and some environmental parameters are monitored by the SCADA system at PUPIN premises, while monitoring of the thermal energy consumption was envisioned to be provided by instalment and integration of additional metering equipment (such as temperature sensors, hot water flow meters and digitalized fuel oil-mazut flow meters/calorimeters). Nevertheless, the proposed interface was designed in flexible and scalable manner so it could be easily extended to encompass subsequently defined metering points.

SCADA View6000 system is capable of integrating various subsystems of automatic control (such as power supply, escalators, fire protection, emergency lighting, etc.) and it provides unique facility management informational environment [14]. It also provides visualization and possibility to archive acquired data and signal values. More precisely, it can control and monitor distributed, heterogeneous subsystems communicating with diversity of field-level devices using different proprietary communication protocols and different types of communication links. Some of the relevant features of the SCADA View6000 are the following:

- acquisition and processing of sensor data (deriving new data from the acquired ones),
- accepting command requests and generating programmed sequences of commands,
- generating events (whenever something irregular was detected, alarm went off, etc.),
- logging sensor readings and derived events (within EVENT/HISTORY database),
- various communication tasks such as selection of the best redundant communication link, and
- presenting relevant information to the operator and other stakeholders.

Configuration data such as, information about field-level devices, their properties, provenance of data, i.e. semantics, are stored in Source/Configuration database, which is read at the SCADA system start-up. It holds semantics of all incoming signals, i.e. it defines a way for processing the incoming raw signals and formulas for determining signal's attributes. After the system start-up, all the configuration parameters as well as the last acquired signal values (from the field) are kept in SCADA shared memory. This shared memory is accessed by different SCADA components as it is presented in Figure 1. Real time data could be the raw sensor readings, derived/aggregated values or set-point data manually defined by the operator. An archiving component stores all the data from the shared memory upon the request (or automatically at regular time intervals) to the Event/History database.

SCADA system at PUPIN campus is processing the raw data which are triggered by the arrival of the data from the communication lines (from remote terminal units - RTU). Both digital and analogue values are monitored by SCADA system. Data read by sensors are converted to the digital value (raw data) and sent to the SCADA along with the information about its source (address). Using the configuration (source) database, SCADA can semantically process (convert or calculate) acquired raw signals according to the information about the source of the data. Finally, the signal attributes are determined after the raw value is processed.

Having in mind the previously listed features of the underlying SCADA system at PUPIN campus, data types which should be managed by the proposed interface were identified as the following:

- Measured data. Data values of measured parameters read from sensors which are already filtered and/or aggregated by the SCADA system. These data are only read. Values of the sensor readings should be accompanied with a unique signal ID indicating that specific reading point.
- Set points. Data values of set points. This data can be only written. Set point values should be accompanied with the corresponding signal ID indicating a specific set point in the system.

3. INTERFACE MODULE DEPLOYMENT

As previously stated, the objective of the proposed interface is to provide the communication with SCADA system. Deployment diagram of interface module is shown in Figure 2. All measured data related to the energy production and consumption are acquired by the interface module and further delivered to the rest of the EMS and energy optimization module in a flexible and secured way. Acquired data are accessible in a web-service based manner supporting different communication protocols (such as FTP, SOAP), and delivered in data polling fashion. In case of PUPIN test-bed platform, communication between SCADA View6000 system and filed-level devices is performed using View6000 legacy protocols and configuration parameters.

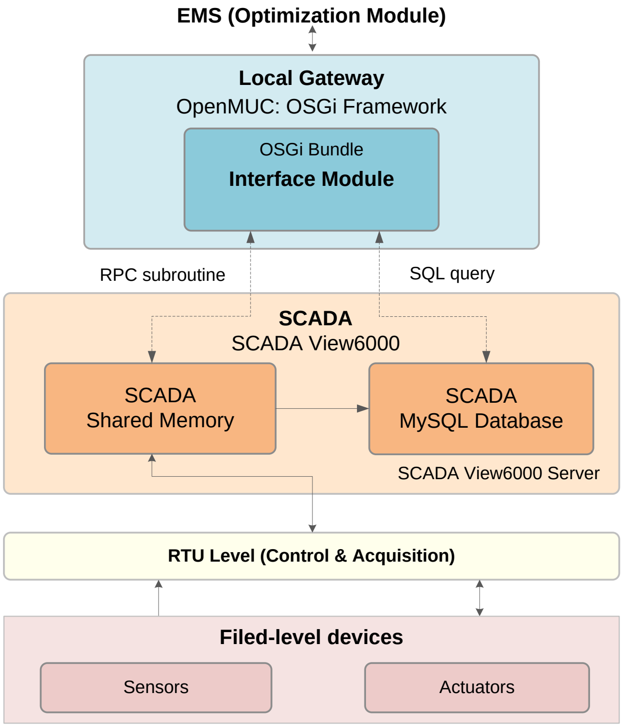

Figure 2: SCADA interface deployment diagram

Independently on the data delivery, energy consumption data monitored through SCADA View6000 system are fetched directly from the SCADA system. In case of data polling, reading of the acquired data is performed by executing the designated "read" method over the available control parameters of the interface module.

At the first place, this is performed by extracting the acquired data directly from the SCADA View6000 History/Event database (through designated remote procedure call - RPC) where all the field-level signals are stored in runtime. By querying the History/Event database (representing the replica of the SCADA memory) any field-level signal value could be acquired and forwarded further to the EMS and energy optimization module at regular time intervals. This requires local communication with SCADA View6000 system deployed at the main PUPIN server. Moreover, the flexible time intervals for fetching the data are supported as well.

When it comes to definition of set-point values and sending the control actions through the SCADA system to the field-level devices, interface was envisioned to support execution of such control actions. Control actions could be performed by executing the designated "write" method over the available control parameters of the related interface module. With respect to the mentioned, control signal could be triggered only within the SCADA View6000 shared memory itself. This further required the development of the interface module capable to trigger the corresponding control signal/set-point value within the SCADA shared memory (again via designated RPC thread). This case required also establishment of the local communication of the interface module with the SCADA system, deployed at the main server at the PUPIN campus.

All the data acquired from the field-level devices/sensors are forwarded from the interface through the Energy Gateway, implemented with the OpenMUC framework [12], to the rest of the EMS. Communication with the Energy Gateway is performed through the OpenMUC data manager using its predefined plug-in interface. Interface module only manages the last values of readings, and therefore there is no information persistence. On the other hand, historical and configuration data are managed by designated sub-component of the Energy Gateway. The deployment of the interface provided the possibility for adjustment of the sampling frequency of monitored parameters in order to meet the requirements of the energy optimization analysis [10],[11]. Furthermore, having in mind that in case of critical infrastructures, open access to the Web is rarely available (due to the security requirements of facility operation), additional constraints should be taken into account such as provision of restricted VLAN for communication between interface module and EMS deployed at the site.

4. DATA SIMULATION AND SCHEDULING

The objective of the data simulation and scheduling component, i.e. data emulator, was to generate and provide the low-level data, such as data related to the energy production and consumption (e.g. electricity, thermal energy, fuel oil, etc.), for the purpose of the verification and testing of the EMS and energy optimization module before the actual deployment within the target infrastructure. In other words, it deals with the simulation and scheduling of the low-level data fed into the system. This includes, first of all, simulation of the data-point values coming from the field-level devices, such as sensors, but also high-level data/parameters such as already filtered and aggregated data-point values coming from SCADA system. By delivering the artificially generated low-level data, early prototyping and calibration of the overall EMS (including its components and energy optimization algorithms) was possible to perform under different use case scenarios. Therefore, data emulator was defined flexible enough to be capable of generating different data types, data formats and protocols. Moreover, it supports flexible time resolution, i.e. granularity of the generated data fed into the EMS.

Data emulator provides, at the first place, a flexible and intuitive way (from the perspective of the end-user, i.e. operator) for definition of various use case scenarios over the chosen time span. This includes the possibility of defining different types of data-points and signals which will be simulated and fed into the EMS (through the Energy Gateway). Low-level data-points and signal values are defined manually. More precisely, the operator have to enter and specify the values of the desired data-points, their specific parameters, time point of generation, etc. This procedure is mainly driven and facilitated by the graphical user interface (GUI) of the data emulator (through corresponding option menus, drop-down lists, default values, etc.) offering some predefined information/parameters for specific data-point types. Such predefined information is currently embedded within the data emulator itself, but it could be also additionally extracted from the EMS central repository/facility model holding various facility/infrastructure related data [15],[16].

Based on the manually defined low-level signals, data emulator provides the possibility to assemble specific signal patterns defined as a set of low-level signals. Such signal patterns could be further utilized (by organizing them over the chosen time span) to assemble the high-level use case scenario which is defined as a sequence of signal patterns. In this way, different, complex use case scenarios could be easily defined by the operator just by definition of the low-level data-point values through the corresponding GUI. Definition of the low-level signal values, signal patterns and high-level use case scenarios is performed off-line. After the definition of the desired use case scenario (stored locally within scenario archive), data emulator provides the possibility to “play” the scenario, i.e. to generate all the low-level signal values according to defined scenario and feed them into the EMS (through the Energy Gateway) for further evaluation.

Data types managed by data emulator are simulated data-point values of measured parameters and set points values. Low-level data-point/signal values generated by data emulator are delivered to the rest of the EMS components and provided in a web-service based manner (supporting different communication protocols such as FTP and SOAP). In terms of generated data delivery, data pushing approach was taken into account. The generated data were fed into the system through the corresponding Energy Gateway (implemented with OpenMUC framework) interface. Communication with EMS Energy Gateway is performed also through the OpenMUC data manager using its predefined plug-in interface.

Logic of the data emulator was wrapped into the bundle-JAR file (based on OSGi specifications [17]) which was deployed under the OpenMUC framework. Data emulator bundle was firing data-point values from previously generated scenario file (for instance, manually specified by the facility operator). It also implements the corresponding Scenario Editor GUI which was developed (in Swing) as a standalone application aimed for user-friendly definition of scenarios through the configuration files that contained all the information needed for scenario simulation. The most important advantage of using the Scenario Editor would be the reusability of data-point value patterns (as well as signal and device patterns), not only within one scenario, but also in other scenarios defined later on, while at the same time providing intuitive and user-friendly approach. In this way, data emulator delivers a convenient way for scenario definition and testing of EMS and energy optimization module.

To have a clear view on the data emulator and its features, integration design and data flow between Scenario Editor GUI component and data emulator bundle component were illustrated in Figure 3. The interaction of these two components (GUI and logic) was envisioned to implement the following action flow:

1) Scenario Editor GUI component is accessible to the operator sitting in front of the EMS cockpit (main user interface) while providing a way for testing EMS in different scenarios (action (1) in diagram: starting the Scenario Editor GUI by the operator)
2) By using the Scenario Editor GUI, operator is capable of defining the data-point values, signals and devices as part of different scenarios (action (2) in diagram: saving the specified scenario into the file by the operator)
3) Desired scenario file is played by activating the data emulator under the OpenMUC framework (action (3) in diagram: starting/activating the data emulator bundle through the OpenMUC framework)
4) Data emulator bundle should "fetch" and read the selected scenario file stored locally or remotely (action (4) in diagram: initialization of bundle with data-point values stored within the scenario file)
5) Data emulator emulates the data-point values according to the scenario file (interaction (5) in diagram: firing the data-point values through the OpenMUC framework)

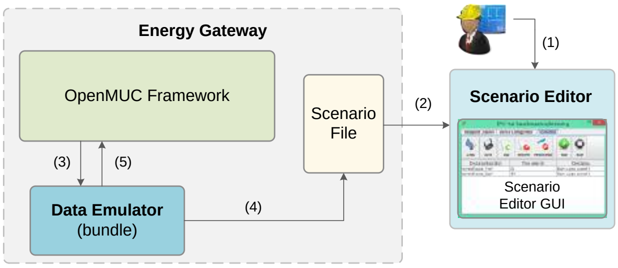

Figure 3: Integration design and data flow of data emulator.

5. SCADA INTERFACE INTEGRATION AND VALIDATION

As previously elaborated, the proposed interface was developed to provide two-way communication with the main BMS system of the target infrastructure. As such, the interface module, together with data simulation and scheduling component, was deployed and validated at PUPIN test-bed platform with multiple energy carrier supply for acquisition of different metering points including electrical and thermal energy production and consumption, and meteorological data, monitored by the SCADA View6000 system.

Interface module was implemented as bundle-JAR file (based on OSGi specifications [17]) which was deployed under the Energy Gateway (OpenMUC framework). For better insight into the data flow, Figure 4 represents simple schematics of interface towards the energy monitoring infrastructure depicting the communication and integration architecture deployed to acquire the designated metering points. As it can be noticed, at the gateway side, the interface forwards all the acquired metering values towards the EMS for further processing.

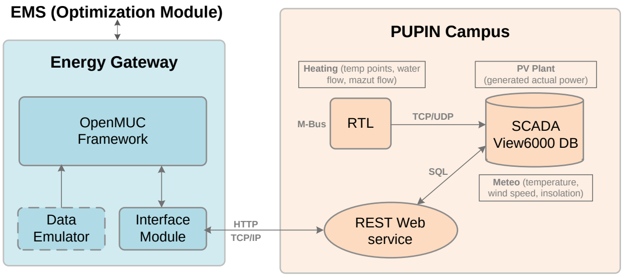

Figure 4. Interface module communication and integration.

At the same time, it communicates with the REST web service (via internet connection using HTTP/TCP/IP) residing at EMS server set at PUPIN premises which performs the acquisition of metering points from SCADA View6000 system supervising the PUPIN premises. Apart from the interface module, data emulator was also deployed under the Energy Gateway to provide data simulation and scheduling required for system validation.

SCADA View6000 system is responsible for supervision of metering points related to the PV plant and energy production (total actual power generated by the plant), heating system, i.e. thermal energy production and consumption (mazut level/consumption, hot water temperature/flow, radiator and air temperature points) and meteorological parameters acquired by PUPIN local meteorological station (outside temperature, wind speed, solar irradiation). As it can be seen from Figure 4, designated RTL unit (ATLAS-MAX/RTL which is an advanced PLC system running on Real-Time Linux) at the field side of PUPIN campus is responsible for acquisition of metering data using M-Bus communication protocol and communication with SCADA View6000 system over TCP/UDP protocol. Additionally, corresponding OpenMUC Gateway parameters have been defined in order to support acquisition of indicated metering points.

For retrieving the metering values stored by the SCADA View6000 system in real time, REST web service was set at designated EMS server. Through the corresponding SQL queries carrying the information about the specific channel/variable IDs, desired metering values were retrieved and forwarded to the interface bundle under the OpenMUC Energy Gateway framework. Finally, the proposed interface was implemented in a scalable and adaptable manner so it could be easily replicated and deployed practically within any complex infrastructure of interest.

6. CONCLUSIONS

The main objective of this paper was development of interface which supported the operation of EMS components by delivering the real-time energy production and consumption data from legacy SCADA systems. This was achieved through the means of the proposed interface module, leveraged upon OpenMUC framework, which offers on-site acquired data primarily towards energy optimization module of EMS which should further take control actions. Therefore, this interface was also responsible for accepting the control decisions, derived from the optimization components, and forwarding them back to the SCADA system for execution via asset control signals. The data are extracted directly from SCADA system real-time database and then offered to other EMS components either directly (if the system is deployed locally) or through the means of web-service technology (REST services). For demonstration of the proposed interface, PUPIN campus was taken as a test-bed platform with SCADA View6000 serving as the main BMS.

Considering that EMS requires involvement of large number of different components and systems, its testing would not be possible on a live system, operating at the demonstration site. Therefore, a software component was developed that provides for a low-level data/signal emulation which would replace signals from the field sensors, monitoring equipment, SCADAs, etc. The developed data emulator offers definition of flexible use case scenarios – including definition of low level signals, signal patterns as well as high-level use case scenarios. Finally, the proposed interface together with the data simulation and scheduling component were developed in such a way to enable bridging between different protocols and technologies. The proposed interface was envisioned as generic and scalable enough to be easily adjusted and applied at any target infrastructure in order to support energy data provision for high-level optimisation and energy management purposes.

ACKNOWLEDGEMENTS

The research presented in this paper is partly financed by the European Union (FP7 EPIC-HUB project, Pr. No: 600067), and partly by the Ministry of Science and Technological Development of Republic of Serbia (SOFIA project, Pr. No: TR-32010).

REFERENCES

[1] Kunold I., Kuller M., Bauer J., Karaoglan N., "A system concept of an energy information system in flats using wireless technologies and smart metering devices," 2011 IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), vol. 2, pp. 812-816, Prague, 15-17.09.2011.
[2] Delimustafic D., Islambegovic J., Aksamovic A., Masic S., "Model of a hybrid renewable energy system: Control, supervision and energy distribution," 2011 IEEE International Symposium on Industrial Electronics (ISIE), pp. 1081=1086, Gdansk, 27-30.06.2011.
[3] Tuladhar A., "Power management of an off-grid PV inverter system with generators and battery banks," 2011 IEEE Power and Energy Society General Meeting, p. 1-5, San Diego, CA, 24-29.07.2011.
[4] Savoie C., Chu V., "Gateway system integrates multiserver/multiclient structure," IEEE Computer Applications in Power, vol. 8, no. 2, pp. 10-14, 1995.
[5] Valsalam S.R., Sathyan A., Shankar S.S., "Distributed SCADA system for optimization of power generation," 2008. INDICON 2008. Annual IEEE India Conference, vol. 1, pp. 212-217, Kanpur, 11-13.12.2008.
[6] Teo C.Y., Tan C.H., Chutatape O., "A prototype for an integrated energy automation system," Conference Record of the 1990 IEEE Industry Applications Society Annual Meeting, vol. 2, pp. 1811-1818, Seattle, USA, 7-12.10.1990.
[7] Ung G.W., Lee P.H., Koh L.M., Choo F.H., "A flexible data acquisition system for energy information," 2010 Conference Proceedings IPEC, pp. 853-857, Singapore, 27-29.10.2010.
[8] Brobak B., Jones L., "SCADA systems for large wind farms: Grid integration and market requirements," 2001 Power Engineering Society Summer Meeting, vol. 1, pp. 20-21, Vancouver, Canada, 15-19.07.2001.
[9] Vankayala V., Vaahedi E., Cave D., Huang M., "Opening Up for Interoperability," IEEE Power and Energy Magazine, vol. 6, no. 2, pp. 61-69, 2008.
[10] Beccuti G., Demiray T., Batic M., Tomasevic N., Vranes S., "Energy Hub Modelling and Optimisation: An Analytical Case-Study", PowerTech 2015 Conference, p. 1-6, Eindhoven 29.06.-02.07.2015.
[11] Batic M., Tomasevic N., Vranes S., "Software Module for Integrated Energy Dispatch Optimization", ICIST 2015, 5th international Conference on Information Society and Technology, pp. 83-88, Kopaonik, 08-11.03.2015.
[12] OPENMUC, Fraunhofer ISE, Germany, www.openmuc.org
[13] SCADA View6000 – Project VIEW Center, Institute Mihajlo Pupin, Serbia, www.view4.rs
[14] Tomasevic N., Konecni G., "Generic message format for integration of SCADA-enabled emergency management systems", 17th Conference and Exhibition YU INFO, pp. 71-76, Kopaonik, 06.03.-09.03.2011.
[15] Tomasevic N., Batic M., Blanes L., Keane M., Vranes S., "Ontology-based Facility Data Model for Energy Management," Advanced Engineering Informatics, vol. 29, no. 4, pp. 971-984, 2015.
[16] Tomasevic N., Batic M., Vranes S., "Ontology-enabled airport energy management," ICIST 2013, 3rd International conference on information society technology, pp. 112-117, Kopaonik, 03-06.03.2013.
[17] OSGi - Open Services Gateway initiative, www.osgi.org