Enabling Adaptivity in IoT-based Smart Grid Architecture

Nenad Petrović and Đorđe Kocić[1]

Abstract –**The increasing usage of electric power in recent years has led to evolution of the existing electric grid infrastructure towards Smart Grid architecture. In this paper, we explore how state-of-art information and communication technologies can be combined to enable adaptivity within the Smart Grid relying on affordable IoT devices. As outcome, we propose an architecture and present some implementation and evaluation aspects.**

Keywords – **IoT, Smart Grid, Edge Computing, semantic technology, data analysis.**

I. Introduction

In the previous century, the usage of electrical power has been one of the main key-enablers of rapid technological progress [1]. However, the demand for electrical energy of today's consumers is becoming much higher, while, on the other side, the availability of non-renewable resources is limited, pushing the traditional energy distribution systems to their limits [1-4]. In such conditions, the quality of the transferred power is dramatically affected that could lead to serious problems and even catastrophic results. Moreover, a constant pressure for switching to renewable, sustainable and cheaper energy resources exists. Therefore, there is a need for evolution of the existing energy distribution systems and increase of their flexibility while making them adaptable.

In recent years, a lot of effort is being put in process of transformation of the existing energy distribution systems, relying on state-of-the-art information and communication technologies, towards the so-called *Smart Grid* infrastructure. Smart Grid is defined as a next generation power grid, implemented as a two-way cyber-physical system with embedded computational intelligence, leveraging the collected information in order to provide clean, safe, secure, reliable, resilient, efficient, economic and sustainable electrical energy to end-users [1-5]. One of its main characteristics is the ability to detect the events that occur anywhere in the grid and react by adopting the corresponding strategy in order to respond the changing demands or recover itself in case of anomalies, in near real-time.

In this paper, it is examined how the synergy of cutting-edge information and communication technologies and paradigms can enable adaptivity within Smart Grid relying on Internet of Things (IoT) devices. As an outcome, we propose an architecture leveraging the mentioned concepts and present proof-of-concept implementation with some evaluation aspects.

II. Background and Related Work

[1]Nenad Petrović and Đorđe Kocić are with the Faculty of Electronic Engineering at University of Nis, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nenad.petrovic@elfak.ni.ac.rs

Architecture model: In literature, many descriptions of Smart Grid infrastructure components and architecture model exist [3-5]. However, some common elements are identified and we summarize them as follows in Table I.

TABLE I
SMART GRID COMPONENTS AND THEIR ROLES

Component	Role
Energy subsystem	Power generation, transmission and distribution
Communication layer	Usage of wired and wireless communication technology to enable information exchange between components
Metering devices	Devices recording electrical and non-electrical measurement values
Computational intelligence	Knowledge extraction from the collected data and decisioning
Applications and services	Various software used by operators or consumers providing visualization, monitoring and/or control

Internet of Things (IoT): IT refers to a system of interconnected devices used in everyday life, residing in our environment with goal to perform the automation of a particular domain – from healthcare and home appliances to military systems. These devices are equipped with different kinds of sensors and modules enabling them to collect certain type of information. Moreover, they can be equipped with actuators in order to be able to affect the environment as a response. In most cases, their processing power is quite limited and they often need to communicate with other devices (or servers) in order to achieve their goal. For communication, a variety of technologies is used, both short- (such as Bluetooth) and long-range (Wi-Fi, 4G). It is identified that IoT has great potential in Smart Grid applications, as measurement and actuation devices have to be distributed throughout residential and industrial objects in order to collect the necessary data and provide the response to the events that have occurred. IoT devices perfectly fit that purpose, considering their small size, affordability and connectivity [2].

Data analysis: In Smart Grids, it is necessary to analyze the enormous amount of data acquired by IoT and metering devices to extract knowledge and meaningful patterns in order to detect or predict occurrence of particular events within the environment and react accordingly. For that purpose, various data mining and machine learning techniques are leveraged. In most of the existing work, the focus of their application is on anomaly detection and load forecasting [3]. In these use cases, clustering, classification and regression are widely used, acting on various measurements beside electrical signals – temperature, weather, rainfall and location data [3-7]. Anomaly detection is of particular importance in this context, as it provides the ability of discovering failures and malfunctions, so Smart Grid can respond to fix them, making them self-healing. In [4], a clustering algorithm together with association rule discovery was applied in order to detect anomalous events, such as overcurrent. On the other side, to optimize the demand scheduling, the accurate energy usage pattern of the consumers is essential. For that reason, the demand forecasting plays an important role. In [6] and [7], regression based on support vector machines was used for load forecasting.

Edge computing: Acquiring the data coming from IoT devices and their sensors is of utmost importance for monitoring and decisioning in Smart Grids. However, in this case, the traditional Cloud computing approach does not give satisfactory results, as offloading the enormous amount of data generated by IoT devices equipped with a variety of sensors to the Cloud for processing would introduce huge latency. On the other side, as Smart Grids grows, the number of connected IoT devices increases, introducing delay even further. As Smart Grid has to act to the environment changes and events in near real-time, such delay is intolerable [8, 9]. The idea of Edge computing is to move the computation and data processing closer to data sources, in order to enable faster response time [8]. For example, in [9], it has been shown that Smart Grid system monitoring performance can be increased up to 10 times by moving the computation closer to the location where the data was generated.

Semantic technology: The role of semantic technology is to encode the meaning of data separately from its content and application code. This way, it is enabled that both machines and people can understand the data, exchange it and perform reasoning. In context of semantic technologies, ontologies are used to describe the shared conceptualization of a particular domain. Semantic descriptions are stored within the triple stores. RDF is often used for the representation of semantic data within triple stores. It consists of classes, their properties and relationships expressed in forms of triplets (*subject, predicate, object*). SPARQL is a language used for querying the RDF semantic triple stores. By executing queries against the triple store, it is possible to retrieve the results that can be further used by reasoning mechanisms in order to infer new knowledge based on the existing facts. In IoT systems, semantic technology is used for various purposes. It is widely adopted in cases when it is needed to achieve interoperability of heterogeneous devices [10]. In [11], a lightweight semantic framework was used for semantic annotation of the results obtained by computer vision algorithms in order to enable reasoning about the events that occurred within IoT-based video surveillance system and act accordingly. In this paper, we want to adapt the similar approach to [11] within Smart Grid architecture.

Domain-specific language (DSL): It is a programming language specialized for solving problems from a particular application domain. If the considered problem belongs to target domain, that problem is solved more conveniently than using general purpose programming languages. Their notation could be textual or visual (within modeling tools). Domain-specific languages are being adopted in IoT systems in order to decrease cognitive load introduced by the device heterogeneity and complexity of the underlying infrastructure. The domain-specific language scripts are further automatically translated to lower-level device-specific commands. For example, in [12], EDL domain specific language is used to describe the experiments carried out using robotic IoT devices. In context of Smart Grids, visual tools based on domain-specific languages would enable much more convenient control and management for operators. This way, the implementation of complex scenarios is enabled by eliminating the need to deal with in-depth implementation details of the involved devices.

III. Implementation Overview

In this section, we propose Smart Grid architecture putting together the previously described technologies and present several aspects of system implementation.

System architecture and working principles: The measurement of electrical quantities is performed by smart devices, referred to as *Smart Meters*. They can either be microcontrollers, low-power single-board computers (such as Raspberry Pi) or even smartphones (as in our case). The advantage of using smartphones is the availability of built-in sensors and inputs (such as audio jack). Moreover, their rechargeable batteries and wireless mobile network availability give the ability to use smartphone devices conveniently even in less accessible areas. After that, the collected data is analyzed relying on data mining and machine learning techniques. The obtained results are semantically annotated, so the semantic reasoning can be performed against them in order to draw conclusion about the events that occurred. According to these results the corresponding actions are taken in order to adapt the Smart Grid to current consumption demands. The adaptation plan can be specified by operators, using a visual modeling tool utilizing a domain-specific notation. Finally, the device-specific commands are generated in order to respond to the changes detected in Smart Grid. The illustration of working principle is given in Fig. 1.

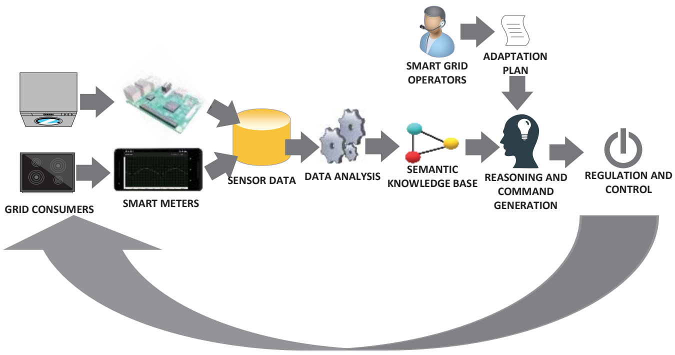

Fig. 1. Overview of automated adaptation process in Smart Grid

Android-based Smart meters: In [13], we have presented a method for acquiring electric measurement utilizing affordable Android-based devices. Voltage and current signals are acquired via voltage and current transformers from the power grid and both converted into voltage signals, which are then scaled down further to audio signal levels using variable resistors. After that signal goes directly into the devices 3.5 mm audio jack. Since many Android based devices support stereo microphone input, it makes them an ideal two-channel measuring platform for power signals. Sound card of the device performs analog to digital (A/D) conversion, so the data can be further processed by standard digital signal processing methods, such as FFT-based algorithms. Moreover, it is useful to also record other data coming from device, such as temperature, timestamp and location. The collected measurements are sent to Edge server, via MQTT (Message Queuing Telemetry Transport)[1], a lightweight, publish-subscribe-based ISO-standard messaging protocol, working on top of TCP/IP. The messages are sent as JSON-encoded string. The smart measurement system is illustrated in Fig. 2.

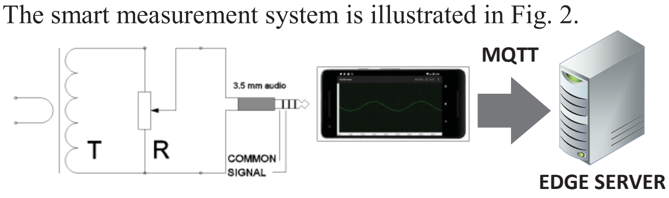

Fig. 2. Android based smart measurement

Data analysis mechanisms: For implementation of data analysis mechanisms, we rely on TensorFlow[2] for Python, an open-source library used for machine learning. We decide to use it, as it supports execution on GPU. Two mechanisms are implemented: 1) anomaly detection based on classification 2) load forecasting based on regression. In the first case, the training data contains voltage and frequency measurements with label (ok/anomaly), while in the second case it consists of average daily consumption (dependent variable) and average temperature (independent variable).

Semantic framework: A domain ontology (illustrated in Fig. 3) is defined in order to semantically annotate the results obtained during the process of data analysis. This way, it is possible to draw conclusion about the events that occurred by executing SPARQL queries and interpreting their results. Different types of events are considered: device failure, voltage anomaly, idle state of the consumer device etc. Moreover, for each of the events, it can be defined which are possible actions that could be taken in order to react to detected events, such as voltage regulation, turning off the device, switching the device to power saving mode.

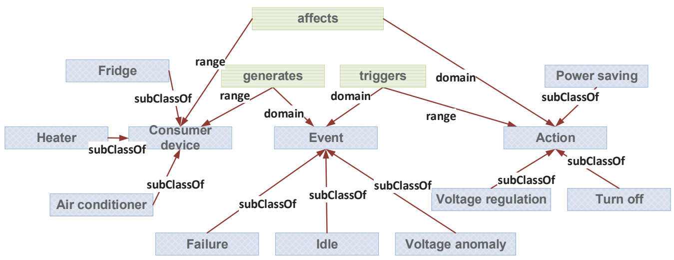

Fig.3. Domain ontology describing control within Smart Grid

Visual modelling tool for grid operators: It was developed using Node-RED[3] as a basis. It is an intuitive and extendable framework that is used for wiring together IoT devices, APIs and online services in novel ways, providing a browser-based editor with drag-and-drop user interface using the wide range of modeling elements (nodes). The domain-specific notation within the tool is described by a metamodel shown in Fig. 4. A model of a modeling language which defines the structure and constraints for a family of models. In this paper, it is used to define a set of actions that need to be taken over the target devices in order to adapt Smart Grid to the environment changes when pre-defined environment conditions (related to the change that occurs) are satisfied. The conditions could be either some specific events or relational expressions with respect to a given pre-defined threshold.

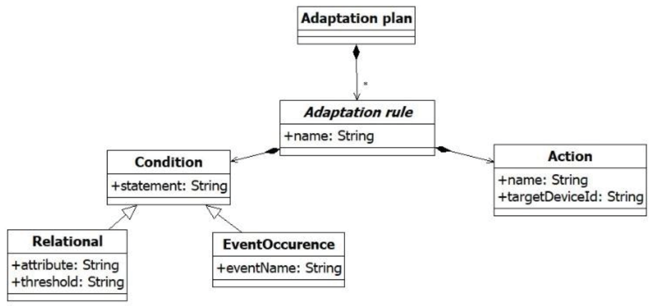

Fig. 4. Adaptation plan metamodel given in UML notation

Reasoning and command generation: It is performed according to the algorithm shown in Listing I as pseudo-code. Each condition from the adaptation plan is translated to SPARQL query and executed against the semantic knowledge base. If it returns results, then the corresponding code is generated and appended to the command script.

LISTING I
CODE GENERATION ALGORITHM

Input: sensor measurements, adaptation plan
Output: commands
Steps:
1. Retrieve all the adaptation rules from the adaptation plan;
2. Analyze sensor data;
3. Semantically annotate results;
4. For each of the adaptation rules
5. If(condition is true)
6. then generate command targeting adaptation_rule.targetDeviceId;
7. end for each
8. end

IV. EVALUATION AND RESULTS

In this section, we present experimental results achieved utilizing the described framework from two different perspectives. First, we consider the accuracy of anomaly detection and load forecasting mechanisms. Moreover, the adaptability responsiveness of the implemented system is analyzed considering the processing time necessary for each step. The evaluation was performed on a server equipped with AMD Ryzen 7 1700X octa-core CPU running at 3.80GHz, 64GB of DDR4 RAM and NVIDIA Quadro P2000 GPU with 4GB of VRAM.

In Table II and Table III, the achieved results using the presented approach for anomaly detection and load forecasting implemented using TensorFlow are given. As it can be noticed

[1] http://mqtt.org/
[2] https://www.tensorflow.org
[3] https://nodered.org/

observing the achieved results, both data analysis mechanisms have satisfactory performance in cases of different training/test set ratios, performing better in case of larger training sets.

TABLE II
ANOMALY DETECTION RESULTS

Training set size	Test set size	Correct/Test size [%]
75	150	89,93
100	150	92,58
125	150	95,17

TABLE III
LOAD FORECASTING RESULTS

Training set size	Test set size	Relative error [%]
75	150	13,51
100	150	12,89
125	150	12,21

In Table IV, an overview of the achieved processing times for data analysis and code generation are shown for various cases of adaptation plan length (in number of rules) and consumer devices involved. Each rule targets a distinct device, while data analysis was performed for measurements collected during 300 seconds. For data analysis, the processing times for anomaly detection considering both the CPU and GPU execution are provided (70% training, 30% test set). According to the results, the time spent for data analysis increases with larger number of devices involved, as more devices generate larger amount of data. Moreover, the time needed for code generation also increases, as number of SPARQL queries that will be executed during the code generation process depends on number of adaptation rules involved. Another observation is that the code generation time for the same number of rules may vary, as it depends on number of adaptation rule conditions that are true, so the commands will be generated only then, as in the second and third case shown in Table IV. Finally, the execution of anomaly detection is up to 3 times faster when executed on GPU, instead of CPU, while the speed-up increases with amount of data, which is beneficial when adaptivity has to be performed in near real-time for huge number of devices.

TABLE IV
PROCESSING TIME OVERVIEW

Number of rules	Data analysis (CPU) [s]	Data analysis (GPU) [s]	Code generation[s]
1	1,74	0,93	1,62
2	2,68	1,06	2,44
2	2,64	1,09	1,91
3	3,71	1,24	3,08

V. CONCLUSION AND FUTURE WORK

In this paper, the enabler technologies for enabling adaptivity in IoT-based Smart Grid architecture were discussed and some implementation and evaluation aspects presented. It can be concluded that IoT-based technology has huge potential in this use case. While the achieved results in case of anomaly detection are comparable to a similar solution [4], the load forecasting performs slightly worse compared to [6, 7]. It can be concluded that more data was needed for training in case of load forecasting. However, our plan is to further work on the implementation, considering the adoption on Big Data technologies, evaluation on larger data sets, optimization of real-time performance, security and evaluation of various implementation variants.

ACKNOWLEDGEMENT

This work is partially supported by grant III44006 by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

[1] A. H. Bagdadee, L. Zhang, "Smart Grid: A Brief Assessment of the Smart Grid Technologies for Modern Power System", Journal of Engineering Technology, vol. 8, no. 1, pp. 122- 142, 2019.
[2] M. Jaradat et al., "The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid", Procedia Computer Science 56 (2015), pp. 592–597, 2015.
[3] B. Rossi, S. Chren, "Smart Grids Data Analysis: A Systematic Mapping Study", ArXiv 2018, pp. 1-26, 2018.
[4] B. Rossi, S. Chren, B. Buhnova, T. Pitner, "Anomaly Detection in Smart Grid Data: An Experience Report", IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016), pp. 1-6, 2016.
[5] H. Shahinzadeh, J. Moradi, G. B. Gharehpetian, H. Nafisi, M. Abedi, "IoT Architecture for Smart Grids", 2019 International Conference on Protection and Automation of Power System (IPAPS), pp. 22-30, 2019.
[6] M. Božić, M. Stojanović, Z. Stajić, "Short-Term Electric Load Forecasting Using Least Square Support Vector Machines", Facta Universitatis, Series: Automatic Control and Robotics vol. 9, no. 1, pp. 141-150, 2010.
[7] A. B. M. S. Ali and S. Azad, "Demand Forecasting in Smart Grid", Green Energy and Technology, pp. 135–150, 2013.
[8] W.Z. Khan et al., Edge computing: A survey, Future Generation Computer Systems (2019), pp. 1-44, 2019.
[9] Y. Huang et al., "An Edge Computing Framework for Real-Time Monitoring in Smart Grid", 2018 IEEE International Conference on Industrial Internet (ICII), pp. 99-108, 2018.
[10] R. Agarwal et al., "Unified IoT ontology to enable interoperability and federation of testbeds", 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 70-75, 2016.
[11] N. Petrovic, "Surveillance System Based on Semantic Video and Audio Annotation Leveraging the Computing Power within the Edge", XIV International SAUM 2018, pp. 281-284, 2018.
[12] K. Kolomvatsos, M. Tsiroukis, S. Hadjiefthymiades, "An Experiment Description Language for Supporting Mobile IoT Applications", FIRE Book, European Commission, River Publishers, pp. 461-486, 2016.
[13] Đ. Kocić, N. Petrović, "Application of Android Based Devices in Analog Electric Signal Measurement", YuInfo 2019, Kopaonik, Serbia, pp. 1-5, 2019.