A WEB-BASED SOLUTION FOR POWER QUALITY DATA MANAGEMENT

Claudiu Popirlan[1], Gabriel Stoian[1], Leonardo Geo Manescu[2], Denisa Rusinaru[2], Marian Ciontu[2], Gabriel Cosmin Buzatu[2], Miron Alba[3], Adrian Cojoaca[3]

[1]Computer Science Department & INCESA Research Hub for Applied Sciences
University of Craiova, Romania
[2]Faculty of Electrical Engineering & INCESA Research Hub for Applied Sciences
University of Craiova, Romania
[3]Oltenia Distribution SA, Craiova, Romania

ABSTRACT

In this paper we present a web-based solution designed to convert (integrating an appropriate plugin) heterogeneous information provided by any type of power acquisition equipment into standard formats as PQDIF (Power Quality Data Interchange Format - IEEE® Std 1159.3-2003 standard). The power grid operators are interested for this kind of applications capable to convert huge volume of heterogeneous information into standard formats in order to be easily processed.

KEYWORDS

Web-Based Solution, Java Enterprise Application, Power Quality, Data Conversion

1. INTRODUCTION

The energy quality data management system provides a deterministic way to take the gross measurements from the points of interest (PoI), and then process them to obtain a unitary representation format, independent of the measuring equipment used. The system also provides users with a simple and easy-to-use interface as well as easy access to collected data for existing power quality analysis applications (such as PQView, see [9], [10]). The architecture of the energy quality management system is shown in Figure 1. This architecture has been described in detail in our previous papers (see [4], [5]).

Points of Interest (PoI) are those locations within the electrical distribution networks where specific equipment for the acquisition of electrical quantities is installed. These data may refer to energy quality if these points of interest are defined as part of an energy quality assessment system / process, but may also refer to other cases: post-incident analysis or simple monitoring of

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP, CMCA - 2018
pp. 85–95, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80907

some electrical quantities. Depending on the installed purchase equipment, data can be collected from them either manually or automatically ready for the next step: loading into SYMPQD.
To reduce the risk of error, users have an extremely convenient way to upload data: the web application "PQDM WebApp". It runs across an application server and provides graphical interface with simple methods to characterize data sets before they are actually loaded into the system. Also, "PQDM WebApp" is responsible for organizing the resulting files according to the needs of their users.

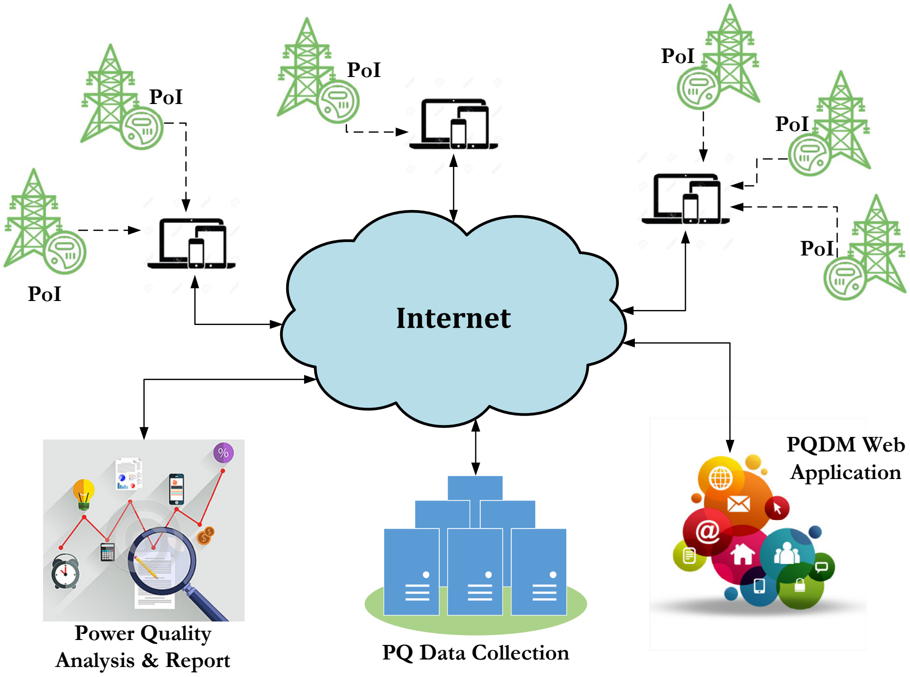


Figure 1. The system architecture of the energy quality management

The user interface is one of the greatest advantages of "PQDM WebApp" because it hides all the details on how to process, internalize and organize the collected data. Beyond this interface, the application is designed to be easily expandable when a new acquisition device is supposed to be supported by SYMPQD. Thus, any such equipment will be analysed to determine its particularities and will then be integrated into SYMPQD by implementing a plugin capable of processing the data collected from it. In this way, the system becomes easy to extend and easy to maintain.

Another strong point of SYMPQD is choosing PQDIF (see [1], [2], [3]) as a format for representing the collected data. PQDIF (Power Quality Data Interchange Format) is a binary file format defined and standardized by the "IEEE 1159.3 PQDIF Task Force" and is the most appropriate way to transfer energy quality data between different software applications of interest. Also, PQDIF is adopted by the overwhelming majority of energy quality analysis and reporting systems, such as PQView (see [9], [10]).

2. WEB-BASED SOLUTION IMPLEMENTATION TECHNOLOGIES

From the point of view of implementation, technologies have followed the natural steps of any software development. We started by performing experiments, measurements, and processing data collected using the Matlab software suite [14]. So we could have access to simple and extremely powerful tools for validating data structures specific to energy systems. Later on for ease of use we opted for a client-server solution where we preferred Java-based technologies [15]. Matlab-based tools have continued to be used as test and validation tools for the resulting files. Using Java, we have benefited from all the benefits of Oracle J2EE [15] that provide robustness, platform independence, efficiency and the ability to use a variety of application servers: Oracle GlassFish [17], Apache Tomcat [16], Eclipse Jetty [18], etc.

PQDM WebApp stores the resulting files in the PQDC (Power Quality Data Collection) location. The PQDC location may be the same as the web application, or it may be different by getting a distributed solution. As a rule, smaller beneficiaries will choose the first option, while large beneficiaries with important IT resources will turn to the distributed version. Regardless of the choice chosen, PQDC offers a unique place where all information on the quality of electricity can be obtained by specialized analysis and reporting software packages.

By choosing PQDIF as the format for the resulting files, we offer our customers the possibility to use a wide range of software solutions for energy quality analysis. Also, by implementing the architecture presented in Figure 1, we significantly reduce the risk of human errors in the processing and organization of collected data, and offer a modern, flexible, scalable, portable and distributed solution.

Implementation technologies and development tools are open, which means we have minimal costs.

2.1. Technical description of PQDM WebApp

At the current stage of the development, the PQDM-WebApp application is developed using Java technologies (https://www.oracle.com/java/index.html), using the NetBeans (https://netbeans.org) development environment (IDE), and is intended to run in the Web.

Because of its functionalities, PQDM-WebApp is classified as a web application, which means that both the server and the client components are working collectively to transcend information from the technical level to a level intended for a much wider audience.

To achieve this goal, java technology was chosen, considering that Oracle J2EE (http://www.oracle.com) offers a lot of advantages: robustness, platform independence, efficiency and the possibility to use a wide range of server applications like Apache Tomcat [16].

The application is based on a three-tier architecture: a presentation tier, a domain logic tier, and a data storage tier. By segregating the application into tiers, offers the advantage to modifying or adding a specific layer, instead of reworking the entire application. The architecture of the web application is given bellow in Figure 2.

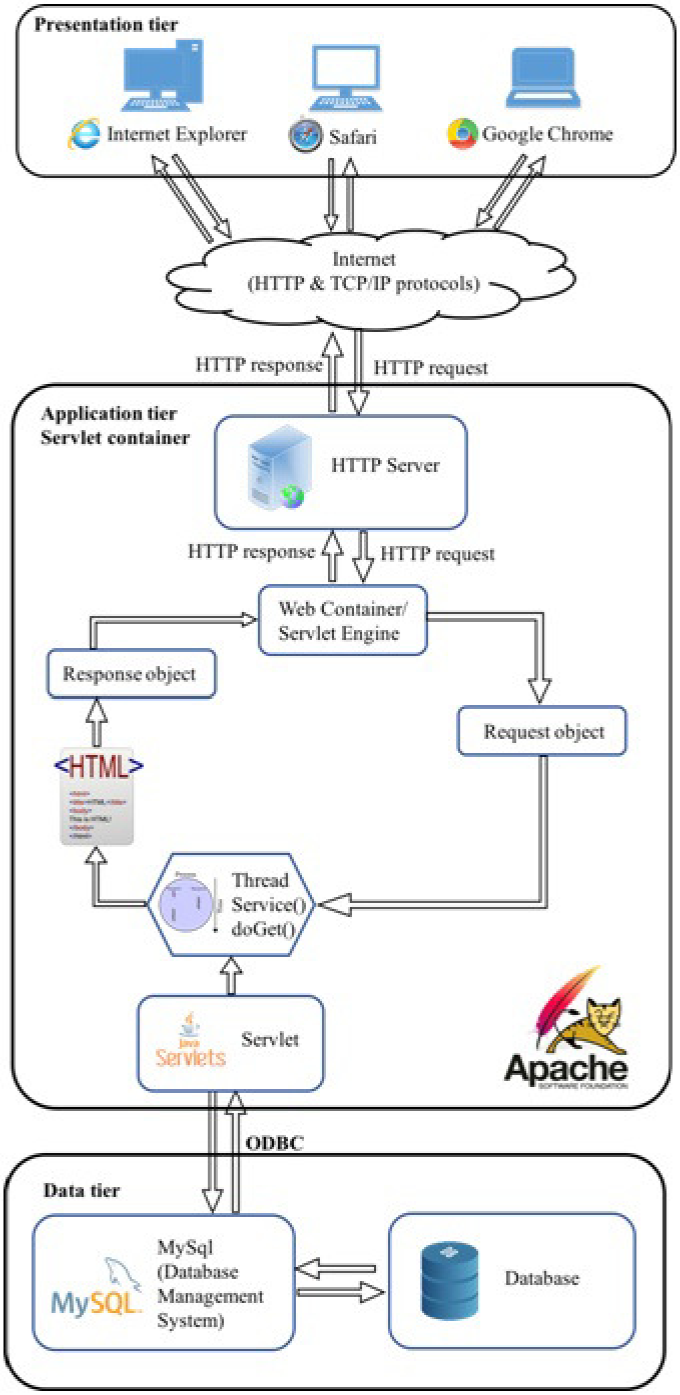


Figure 2. The PQDM-WebApp architecture

This three-tire client-server software architecture pattern with its functionalities, and their subcomponents determined by the chosen technologies are as following:

1. Presentation tier: The topmost level application is the user interface. The main function of the interface is to translate tasks and results the user can understand and access directly

(such as a web page, or an operating system's GUI). In this case, the tire is composed of the web browsers: Internet Explorer, Safari, Google Chrome, etc.

2. Application tier: This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers. By choosing to use java technology, a suited environment to run the code is Apache Tomcat, which is an open-source Java Servlet Container.

3. Data tier: Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

A short description in a deductive way of the terms that are further used, is given. Abstract data type is a data type defined by the operations that can be applied on them, and are implemented through a class, where operations are called methods or functions. An object is an instance of a class and represents what actually is running on a computer. A container is an object which contains multiple objects.

Apache Tomcat server operates according to the principle of request-response model. A request is send when from the client level is demanded a web page that exists in the context of the server. By executing the corresponding Java Servlet class, a response is generated, which consists in a HTML document (HyperText Markup Language).

Java Virtual Machine is created when the Apache Tomcat server is started and destroyed when exited. Its main purpose is to enable a computer to run a Java program.

Inside the Java Virtual Machine is running a single server which represents the entire Tomcat instance. Its purpose is to manage the life cycle of its contained services. A server can contain one or more Services, where each Service represents the set of request processing components within Tomcat.

The service object is a structural element that combines one or more Connector components and a single Catalina servlet Engine element.

Giving the dynamic nature of the application, Tomcat is configured to work in conjunction mode with a separate HTTP web server, therefore two connectors are used:

- The HTTP/1.1 Connector manages the communication between the client and the web server through the HTTP protocol (HyperText Transfer Protocol), which listen on port 8080. If a HTTP request is needed to be routed to the Tomcat container for processing, the web server will communicate this request using the AJP protocol (Apache JServer Protocol), which is a binary protocol more efficient than the text-based HTTP.

- The AJP/1.3 Connector, placed inside the service, is listening on port 8009 for requests, and translate them into a request object which Catalina engine can process.

The Catalina container's components that are responsible for processing those requests and generate response objects are: the engine, virtual host, and context.

The Host component is a container for web application(s) or context(s), and it is defined by two fundamental concepts:

(1) Domain name: this is the value expected to be sent by the client browser (http://localhost:8084/PQDM-WebApp);

(2) Application base name: in this folder is located the context that will be deployed to this host.
The context or web application (PQDM-WebApp) is the location where the application specific servlet and JSP (JavaServer Page) and their associated files live.

A Wrapper object is a child of the context container and represents an individual servlet (or a JSP file converted to servlet).

In the life cycle of a servlet, three methods are fundamental: *init()*, *service()* and *destroy()*, which are implemented by every servlet and are invoked at specific times by the server. The course of action of these three methods are in the following order:

1. During initialization stage of the servlet life cycle, the web container initializes the servlet instance by calling the *init()* method, passing an object implementing the *javax.servlet.ServletConfig* (https://docs.oracle.com) interface. This configuration object allows the servlet to access name-value initialization parameters from the web application.

2. After initialization, the servlet instance can service client request. Each request is serviced in its own separate thread. The web container calls the *service()* method of the servlet for each request. The *service()* method determines the kind of request being made and dispatches it to the appropriate methods: *doGet()*, *doPost()*, *doPut()*, *doDelete()*, and so on; according to the HTTP request.

3. Finally, the web container calls the *destroy()* method that takes the servlet out of service. The *destroy()* method, like *init()*, is called only once in the lifecycle of a servlet.

A thread represents the actual execution of a servlet class deployed by the processor to generate a response to the request. During the class execution, in order to generate the response object, it has to interact with the data tire passing a set of SQL (Structured Query Language) queries to the Database Management System (DBMS). The access of DBMS is realized through an ODBC (Open Database Connectivity).

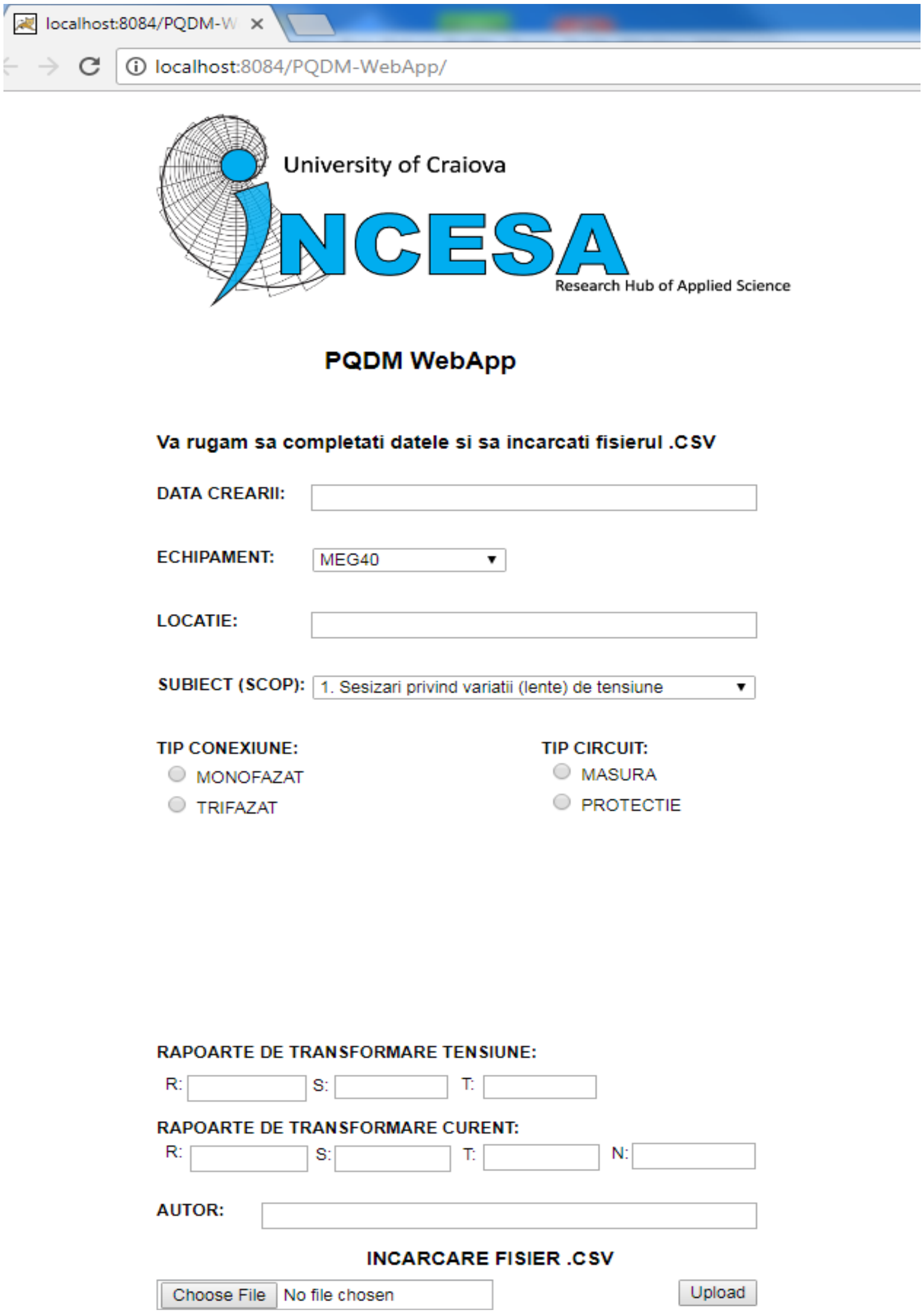
The DBMS used to store the data collected from the power analyser's equipment is MySQL (http://www.oracle.com), which is an open-source relational database management system.

2.2. PQDM-WebApp in running mode. Utilisation

The user accesses the web application PQDM-WebApp by entering the address provided by the network administrator in the browser according to the application server it will run.

For example, we used the server on a local machine: http://localhost:8080/PQDM-WebApp/
The PQDM-WebApp application receives the file for the selected device along with the user metadata's and returns the PQDIF (http://grouper.ieee.org/groups/1159/3/).

The user must introduce the metadata's in PQDM-WebApp interface and select the corresponding file for the selected power equipment as is shown in Figure 3.



PQDM WebApp

Va rugam sa completati datele si sa incarcati fisierul .CSV

DATA CREARII:

ECHIPAMENT: MEG40 ▾

LOCATIE:

SUBIECT (SCOP): 1. Sesizari privind variatii (lente) de tensiune ▾

TIP CONEXIUNE:
○ MONOFAZAT
○ TRIFAZAT

TIP CIRCUIT:
○ MASURA
○ PROTECTIE

RAPOARTE DE TRANSFORMARE TENSIUNE:
R: S: T:

RAPOARTE DE TRANSFORMARE CURENT:
R: S: T: N:

AUTOR:

INCARCARE FISIER .CSV

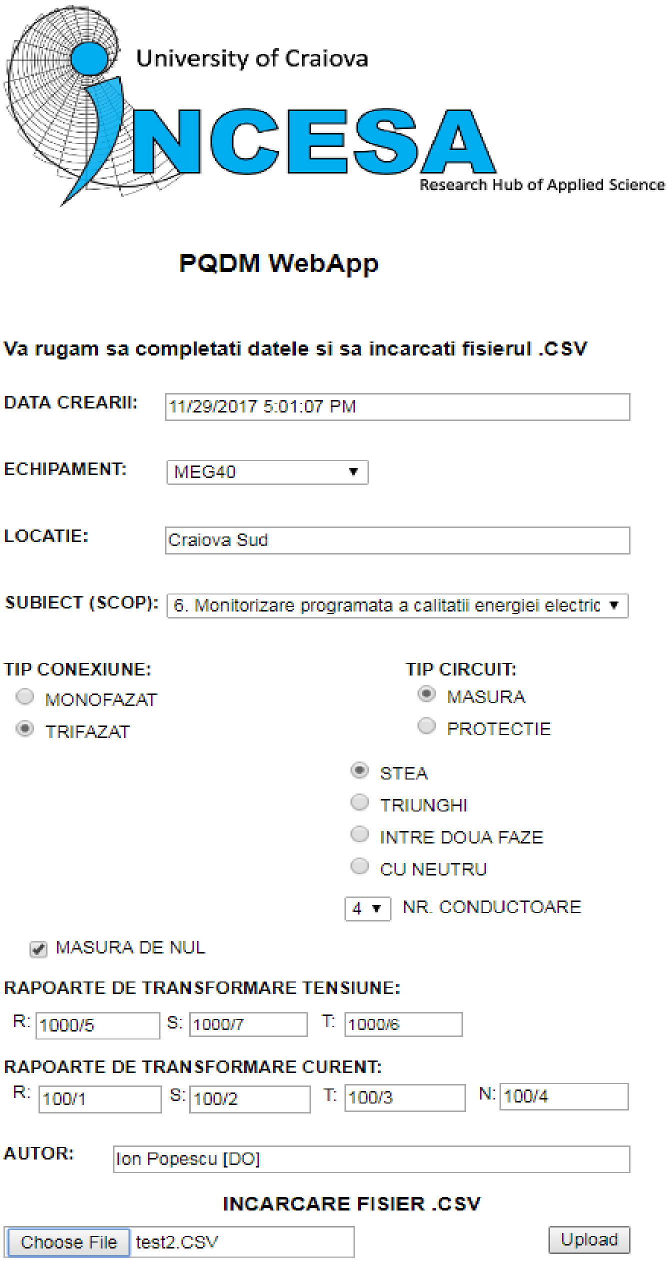
Choose File No file chosen Upload

Figure 3. PQDM-WebApp User Interface

Consistent with the structure of the PQDIF file that will be generated through the PQDM-WebApp application, the user will fill in the information in the web interface formats. The details of the completion of the information will be presented in the next section.

2.3. Case Study

An analysis was carried out for a MEG40 device (http://www.e-mega.cz/meg-40) together with the following user input information's: Version Info, File Name, Creation, Subject, Author, Application, Copyright, Trademark, Notes, Language, Owner.



PQDM WebApp

Va rugam sa completati datele si sa incarcati fisierul .CSV

DATA CREARII:	11/29/2017 5:01:07 PM
ECHIPAMENT:	MEG40 ▼
LOCATIE:	Craiova Sud
SUBIECT (SCOP):	6. Monitorizare programata a calitatii energiei electric ▼

TIP CONEXIUNE:
○ MONOFAZAT
● TRIFAZAT

TIP CIRCUIT:
● MASURA
○ PROTECTIE

● STEA
○ TRIUNGHI
○ INTRE DOUA FAZE
○ CU NEUTRU

4 ▼ NR. CONDUCTOARE

☑ MASURA DE NUL

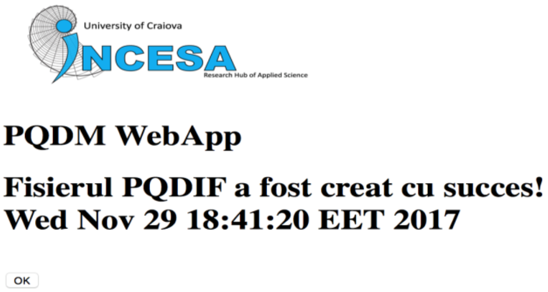
RAPOARTE DE TRANSFORMARE TENSIUNE:
R: 1000/5 S: 1000/7 T: 1000/6

RAPOARTE DE TRANSFORMARE CURENT:
R: 100/1 S: 100/2 T: 100/3 N: 100/4

AUTOR: Ion Popescu [DO]

INCARCARE FISIER .CSV
Choose File test2.CSV Upload

Figure 4. PQDM-WebApp utilization



PQDM WebApp

Fisierul PQDIF a fost creat cu succes!
Wed Nov 29 18:41:20 EET 2017

Figure 5. PQDM-WebApp – PQDIF success file creation

After the metadata and the .CSV file for MEG40 are inserted into the web interface, the PQDM-WebApp creates the corresponding PQDIF file, as shown in Figure 5.

3. CONCLUSIONS

In this paper a web-based solution, which manage and convert data regarding the power equipment's and the power quality in the power distribution networks, was presented.

During the research, we were forced to automate the common process in order to minimize the time spent in transfer/copying data. As a future work we try to improve this automation in order to optimize the whole process.

This web-based solution integrates several data sources and analysis tools. It associates the capabilities of the commercial software platforms with those of an in-house designed software named PQDM-WebApp.

This application allowing us to extend the capabilities of the commercial software packages and to develop a complex data management of the power grid networks based on its typical operation regimes, as well as on the real historical evolution of its electric values. The management and the conversion modules of web application enable the system configuring, as well as report generation and data visualization for a heterogeneous large volume of data. The architecture is flexible, highly portable, distributed and platform independent.

ACKNOWLEDGEMENTS

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-BG-2016-0269, within PNCDI III".

REFERENCES

[1] IEEE, 2008, IEEE Std 1159.3 Recommended Practice for the Transfer of Power Quality Data.

[2] http://grouper.ieee.org/groups/1159/3/ (Accesed date: March 2018)

[3] D. Sabin, A. R. Dettloff, P. Golden, “Automatic Subtransmission Fault Location System using Power Quality Monitors”, Transmission and Distribution Conference and Exposition (T&D), 2016, DOI: 10.1109/TDC.2016.7519887

[4] L.G. Manescu, D. Rusinaru, C. Popirlan, G. Stoian, M. Ciontu, G.C. Buzatu, M. Alba, A. Cojoaca, “Complex Software System for Data Management and Analysis of Power Distribution Grids”, MCSI 2017, Corfu, Greece, mcsi-142.

[5] D. Rusinaru, L.G. Manescu, M. Ciontu, P.M. Mircea, G.C. Buzatu, G. Stoian, C. Popirlan, T. Vilceanu, A. Negoita, M. Alba, “Powering knowledge transfer at INCESA-BRIDGE projects”, Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on, 25-27 May 2017, Brasov, Romania, http://ieeexplore.ieee.org/abstract/document/7974969/, pp. 190-195, DOI: 10.1109/ OPTIM.2017.7974969, Publisher IEEE

[6] P. Martí, M. Velasco, J. Torres-Martínez, J. Miret and M. Castilla, "Reactive power control for loss minimization in low-voltage distributed generation systems," 2016 12th IEEE International Conference on Control and Automation (ICCA), Kathmandu, 2016, pp. 371-376.

[7] L. G. Manescu and D. Ruşinaru, "Loss based performance index for the reactive power control," Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference on, Brasov, 2012, pp. 307-312.

[8] L. Bam, W. Jewell, “Review: power system analysis software tools”, Power Engineering Society General Meet., 2005. IEEE, vol 1, pp. 139–44.

[9] www.pqview.com (Accesed date: May 2018)

[10] PQView User Manual and Quick Start Guide, Electrotek Concepts, Inc., 2013.

[11] Rusinaru, D., Manescu, L.G., Merfu, M., Postolache, P., “Power quality general levels in distribution networks”, 16th International Conference on Harmonics and Quality of Power (ICHQP), 2014 IEEE, Bucharest, Romania, 25-28 May 2014, pp. 58 – 62.

[12] C.I. Popirlan, “A solution based on intelligent software agents to improve the data searching in the contact centers”, 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, IEEE ITAIC 2011, Chongqing, China, ISBN(Print) 978-1-4244-8623-6, Vol. 2, pp. 1-5.

[13] G. Stoian, C.I. Popirlan, “Using Mobile Agents for Handoff Tuning in Mobile WiMAX Networks”, Annals of the University of Craiova - Mathematics and Computer Science Series, ISSN 1223-6934, Vol. 37, No. 3, 2010, pp. 22-28

[14] https://www.mathworks.com/products/matlab.html

[15] https://www.oracle.com/java/technologies/index.html

[16] http://tomcat.apache.org

[17] http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html

[18] https://www.eclipse.org/jetty/

[19] http://www.e-mega.cz/meg-40

AUTHORS

INCESA (Research Hub for Applied Sciences) is intended to be an inspiring co-working space for driving effective innovation for the benefit of ever wider communities. www.incesa.ro

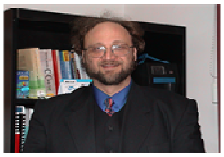
Claudiu Popirlan and Gabriel Stoian are members at Computer Science Department, University of Craiova and researchers at Laboratory of Formal Intelligence Integration in Analysis, Simulation, Development, Testing and Certification of Computation Infrastructures, INCESA (http://incesa.ro/#/computer_science/computation_infrastructure).

Leonardo Geo Manescu, Denisa Rusinaru, Marian Ciontu and Gabriel Cosmin Buzatu are members at Faculty of Electrical Engineering, University of Craiova and researchers at Laboratory of Innovative Techniques and Processes in Complex Electromechanical Systems, INCESA (http://incesa.ro/#/electrical_engeneering/smart_grids)

Miron Alba and Adrian Cojoaca are members of the company Oltenia Distribution SA (https://distributieoltenia.ro/en) and have a great experience in developing applications in the field of electrical engineering.

The authors have numerous scientific papers in the fields of computer science and electrical engineering and many participations in international conferences around the world.

Claudiu Popirlan



Gabriel Stoian

Leonardo Geo Manescu

Marian Ciontu

Miron Alba

Adrian Cojoaca