APPLICATION DEVELOPMENT TO CONVERT HETEROGENEOUS INFORMATION INTO PQDIF (POWER QUALITY DATA INTERCHANGE FORMAT)

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 is presented a web-based application development, used to convert heterogeneous information provided by MEG40 (power acquisition equipment) into standard formats as PQDIF (Power Quality Data Interchange Format - IEEE® Std 1159.3-2003 standard). The web-based solution proposed is able 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 is able to provide a method that can take measurements from the point of interest and transform them into a standard representation format. The representation obtained does not depend on the equipment used to make measurements. The data management system has a friendly interface and provides 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, presented in detail in our previous papers (see [4], [5]) is illustrated in Figure 1.

The locations within the electrical distribution networks are named Points of Interest (PoI) and specific equipment used for electrical quantities acquisition is installed in these locations. If the points of interest are considered parts of an energy quality assessment system process, then the equipment quantify the energy quality. The data obtained from this equipment may refer to: energy quality, post-incident analysis or simple monitoring of 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.

The web application named "PQDM Web App" is used by the users to upload data. This application is used to reduce the risk of errors because it provides a user friendly graphical interface that uses methods to characterize data sets before they are actually loaded into the system. The application "PQDM Web App" also organize the resulting files according to the needs of their users.

DOI: 10.5121/ijcsit.2018.10401

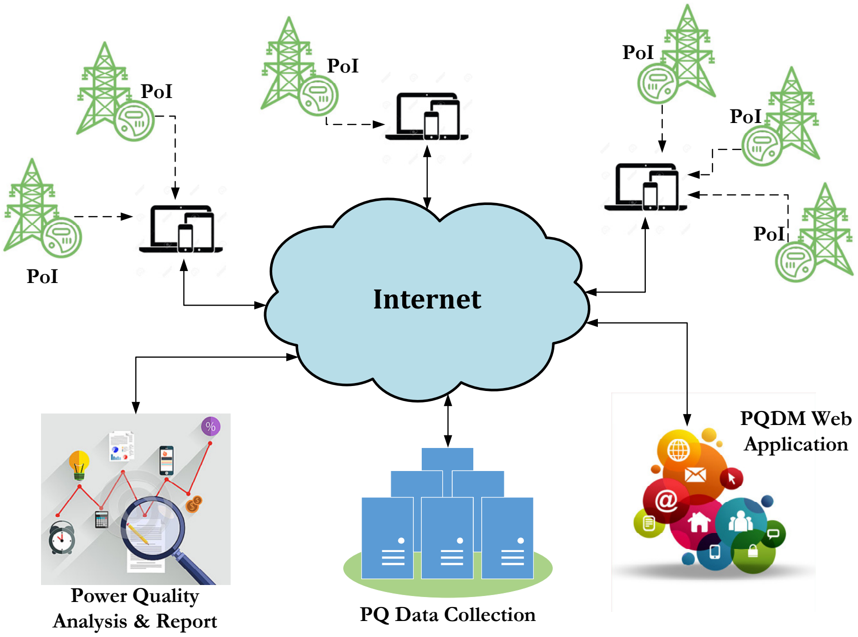


Figure1. The system architecture of the energy quality management

One of the biggest advantage of "PQDM WebApp" is its user interface, all the details about data process, internalize and organize are hidden. Due its design the application can be easily expanded 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.

The format used for representing collected data is PQDIF (see [1], [2], [3]) and so this is an important advantage of the application SYMPQD. 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. Matlab software [4] was first use to perform experiments, to take measurements and to process collected data. This software allowed us to use its tools for data structure validation in our energy system. Later on we opted for a client-server solution where we preferred Java-based technologies [15], but we continued to use Matlab tools for tests and validation 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. The resulting files of our application PQDM WebApp are stored in the PQDC (Power Quality Data Collection) location. Smaller beneficiaries of the application will choose to store their files in the same location as the application, while large beneficiaries with important IT resources will turn to the distributed version. So our application is suitable for a distributed solution. No matter the chosen storage solution, the application PQDC offers a unique place where all information on the quality of electricity can be obtained by specialized analysis and reporting software packages. Because we use 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 Web App

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.

The server and the client components of PQDM-WebApp are working collectively to transcend information from the technical level to a level intended for a much wider audience, and due to these functionalities, our application is a web application.

In order to achieve the proposed goal, we chooses java technology, knowing the advantages offered by Oracle J2EE (http://www.oracle.com): robustness and efficiency, platform independence and the possibility to use a wide range of server applications like Apache Tomcat [16].

The architecture of our application consists in three-tier (presentation tier, logic domain tier and data storage tier) which allows us to easily modify the application by adding other specific tier, this way we do not have to rework on the entire application. This tier structure architecture of our web application is presented in Figure 2.

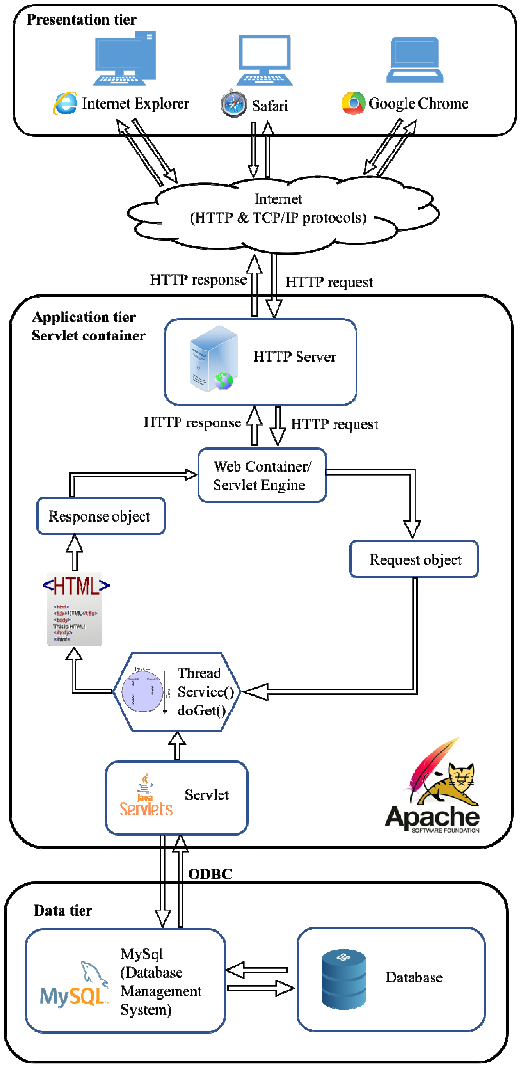


Figure 2. The PQDM-WebApp architecture

The functionality and the components of the three-tire client-server software architecture are determinate by the technologies used in software development and they can be described as following:

1. Presentation tier: The topmost level application is the user interface. The main functionality of the interface is to translate tasks and results so that the user can understand and access directly (such as a web page, or an operating system's GUI). For our application, the presentation tire is composed of 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: This tire allows the user to store and retrieved data from a database or file system. After the information is retrieved, it is passed back to the application tier for processing, and then, eventually, the results are return 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. Our application are use two connectors, that are determinate by the dynamic nature of the application and by the Tomcat configuration to work in conjunction mode with a separate HTTP web server:

- The HTTP/1.1 Connector which 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, that is placed inside the service, is listening on port 8009 for requests, and translates them into a request object which Catalina engine can process.

Responsible for requests' processing and response's generation are the Catalina container's components: engine, virtual host and context. The Host component is a container for web application or context. It isdefined by two fundamental concepts:

(1) The domain name sent by the client to the browser (http://localhost:8084/PQDM-WebApp);

(2) The application base name: folder containing the context that will be deployed to this host.

The web application PQDM-WebApp (the context) share the same location as the application specific servlet, JSP (JavaServer Page) and their associated files.

A Wrapper object is a child of the context container and represents an individual servlet (or a JSP file converted to servlet). For a wrapper we identify three fundamental methods:

init(), *service()* and *destroy()*. These methods are implemented by every servlet and they are used by the server at specific times. 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. Every request is processed in a separate thread. For each request, the *service()* method is called by the servlet web container. 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 and it takes the servlet out of service. The *destroy()* method, like *init()*, is called only once in the lifecycle of a servlet.

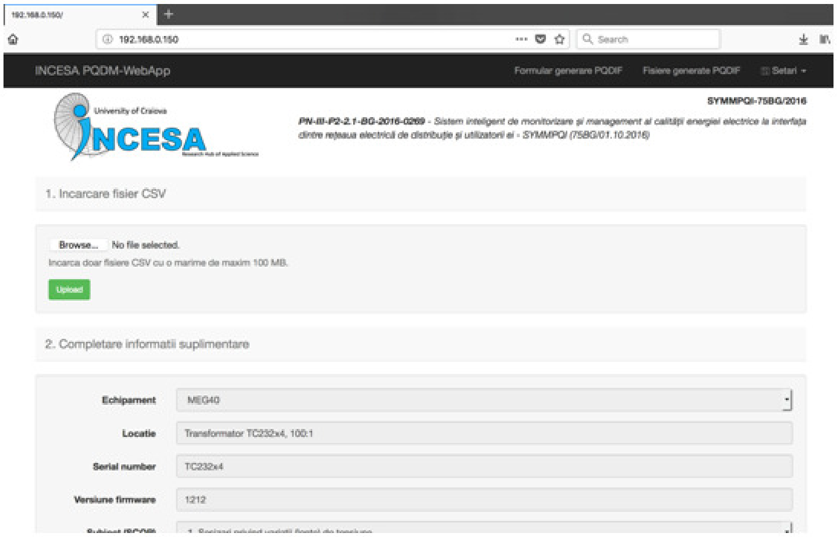
A thread represents the actual execution of a servlet class deployed by the processor to generate a response to the request. During the class execution, in order to generate the response object, it has to interact with the data tire passing a set of SQL (Structured Query Language) queries to the Database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

In our application we use an open-source relational database management system, MySQL (http://www.oracle.com), as a DBMS to store all data that are collected from the power analyser's equipment.

2.2. PQDM-Web App in running mode. Utilisation

The web application PQDM-WebApp can be easily accessed by the user only by entering in the browser the address of the server application provided by the network administrator.In order to exemplify, we used a local machine for the server: http://localhost:8080/PQDM-WebApp/

The metadata are introduced in the application interface by the user (Figure 3), who also select corresponding file for the selected power equipment. The files from the selected drive and the user's metadata are sent to the application PQDM-WebApp which returns the PQDIF (http://grouper.ieee.org/groups/1159/3/).



SYMMPQI-75BG/2016

PN-III-P2-2.1-BG-2016-0269 - Sistem inteligent de monitorizare și management al calității energiei electrice la interfața dintre rețeaua electrică de distribuție și utilizatorii ei - SYMMPQI (75BG/01.10.2016)

1. Incarcare fisier CSV

No file selected.

Incarca doar fisiere CSV cu o marime de maxim 100 MB.

2. Completare informatii suplimentare

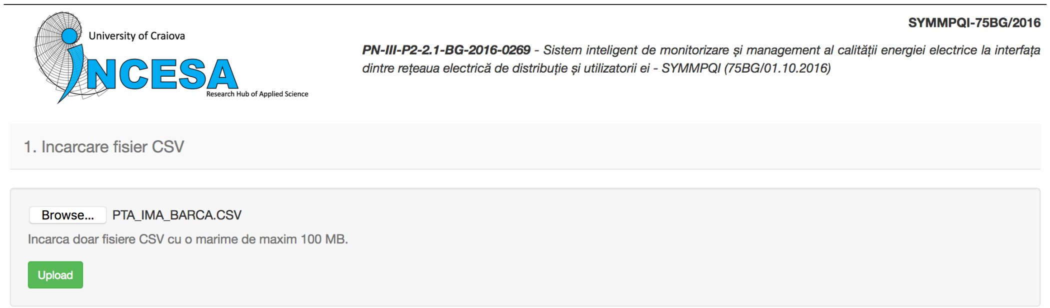
Echipament	MEG40
Locatie	Transformator TC232x4, 100:1
Serial number	TC232x4
Versiune firmware	1212
Subiect (PQDIF)	1. Sesizari privind variatii lente de tensiune

Figure 3. PQDM-WebApp User Interface

The user fills all necessary data in the application interface so that PQDM-WebApp generates the PQDIF file. In the next section the details about information needed to be completed in the interface will be presented.

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.



SYMMPQI-75BG/2016

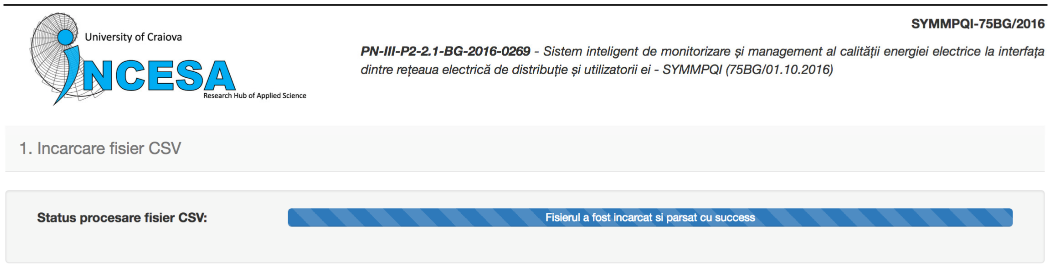
PN-III-P2-2.1-BG-2016-0269 - Sistem inteligent de monitorizare și management al calității energiei electrice la interfața dintre rețeaua electrică de distribuție și utilizatorii ei - SYMMPQI (75BG/01.10.2016)

1. Incarcare fisier CSV

PTA_IMA_BARCA.CSV

Incarca doar fisiere CSV cu o marime de maxim 100 MB.

Figure 4. PQDM-WebApp utilisation



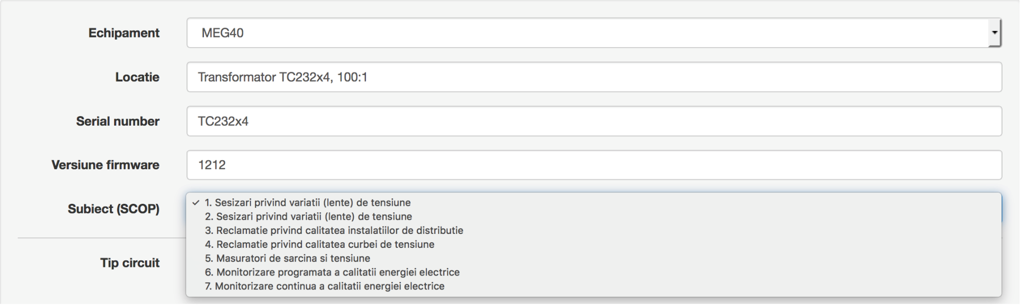
SYMMPQI-75BG/2016

PN-III-P2-2.1-BG-2016-0269 - Sistem inteligent de monitorizare și management al calității energiei electrice la interfața dintre rețeaua electrică de distribuție și utilizatorii ei - SYMMPQI (75BG/01.10.2016)

1. Incarcare fisier CSV

Status procesare fisier CSV: Fisierul a fost incarcat si parsat cu success

Figure 5. PQDM-WebApp utilisation



Echipament MEG40

Locatie Transformator TC232x4, 100:1

Serial number TC232x4

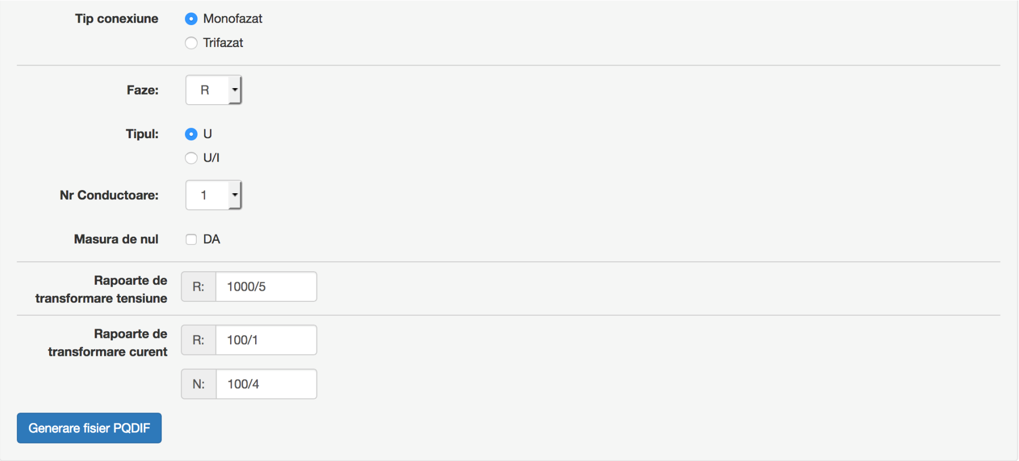
Versiune firmware 1212

Subiect (SCOP)

✓ 1. Sesizari privind variatii (lente) de tensiune
2. Sesizari privind variatii (lente) de tensiune
3. Reclamatie privind calitatea instalatiilor de distributie
4. Reclamatie privind calitatea curbei de tensiune
5. Masuratori de sarcina si tensiune
6. Monitorizare programata a calitatii energiei electrice
7. Monitorizare continua a calitatii energiei electrice

Tip circuit

Figure 6. PQDM-WebApp utilisation



Tip conexiune ● Monofazat ○ Trifazat

Faze: R

Tipul: ● U ○ U/I

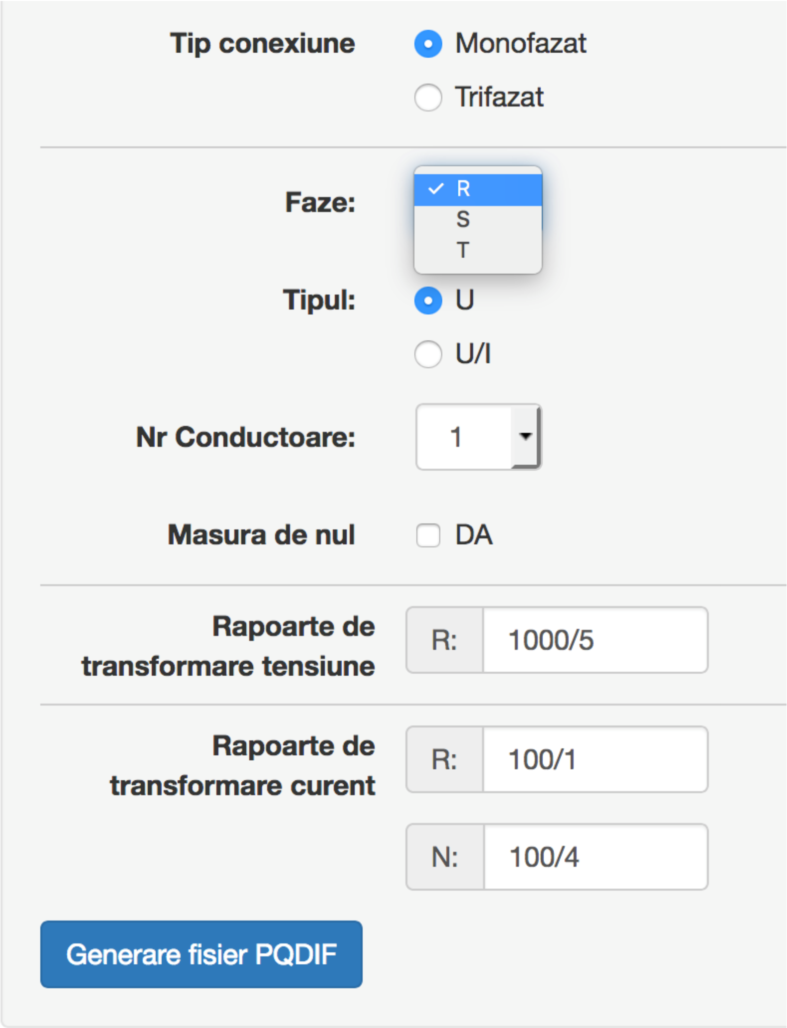
Nr Conductoare: 1

Masura de nul □ DA

Rapoarte de transformare tensiune R: 1000/5

Rapoarte de transformare curent R: 100/1 N: 100/4

Figure 7. PQDM-WebApp utilisation



Tip conexiune: ● Monofazat ○ Trifazat

Faze: ✓ R, S, T

Tipul: ● U ○ U/I

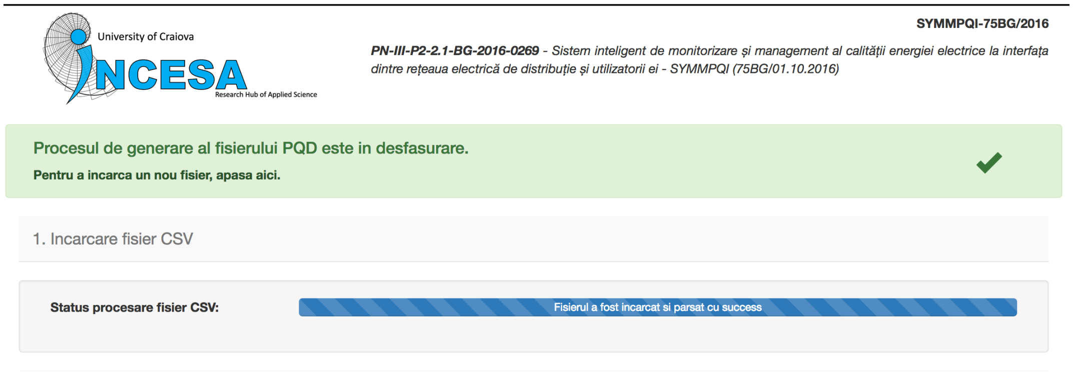
Nr Conductoare: 1

Masura de nul: □ DA

Rapoarte de transformare tensiune: R: 1000/5

Rapoarte de transformare curent: R: 100/1; N: 100/4

Figure 8. PQDM-WebApp utilisation



SYMMPQI-75BG/2016

PN-III-P2-2.1-BG-2016-0269 *- Sistem inteligent de monitorizare și management al calității energiei electrice la interfața dintre rețeaua electrică de distribuție și utilizatorii ei - SYMMPQI (75BG/01.10.2016)*

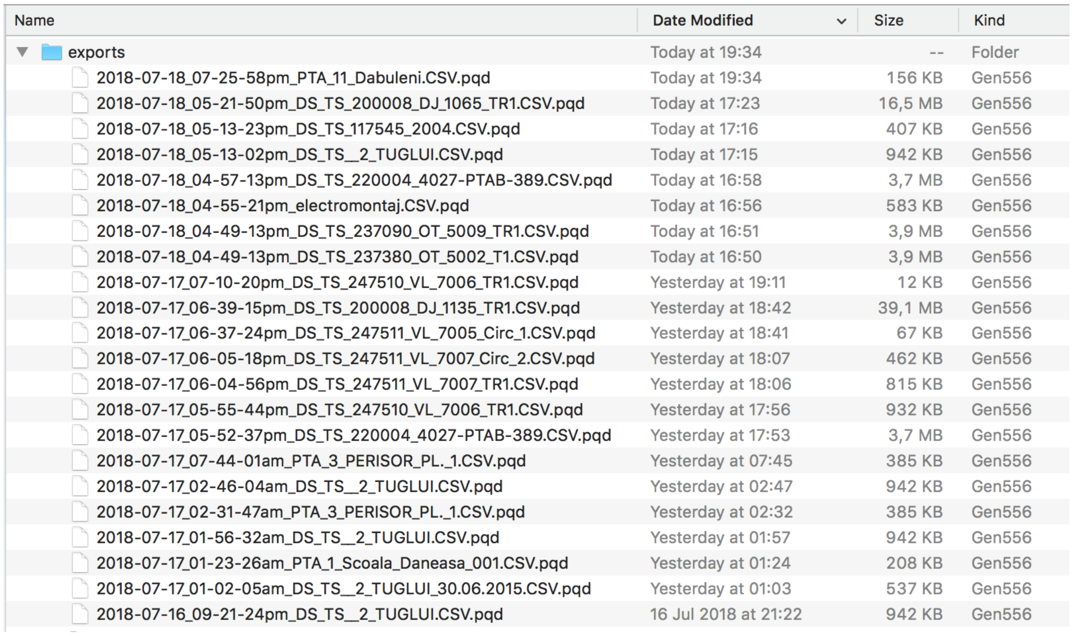
Procesul de generare al fisierului PQD este in desfasurare.

Pentru a incarca un nou fisier, apasa aici. ✓

1. Incarcare fisier CSV

Status procesare fisier CSV: Fisierul a fost incarcat si parsat cu success

Figure 9. PQDM-WebApp – PQDIF success file creation



Name	Date Modified	Size	Kind
▼ exports	Today at 19:34	--	Folder
2018-07-18_07-25-58pm_PTA_11_Dabuleni.CSV.pqd	Today at 19:34	156 KB	Gen556
2018-07-18_05-21-50pm_DS_TS_200008_DJ_1065_TR1.CSV.pqd	Today at 17:23	16,5 MB	Gen556
2018-07-18_05-13-23pm_DS_TS_117545_2004.CSV.pqd	Today at 17:16	407 KB	Gen556
2018-07-18_05-13-02pm_DS_TS__2_TUGLUI.CSV.pqd	Today at 17:15	942 KB	Gen556
2018-07-18_04-57-13pm_DS_TS_220004_4027-PTAB-389.CSV.pqd	Today at 16:58	3,7 MB	Gen556
2018-07-18_04-55-21pm_electromontaj.CSV.pqd	Today at 16:56	583 KB	Gen556
2018-07-18_04-49-13pm_DS_TS_237090_OT_5009_TR1.CSV.pqd	Today at 16:51	3,9 MB	Gen556
2018-07-18_04-49-13pm_DS_TS_237380_OT_5002_T1.CSV.pqd	Today at 16:50	3,9 MB	Gen556
2018-07-17_07-10-20pm_DS_TS_247510_VL_7006_TR1.CSV.pqd	Yesterday at 19:11	12 KB	Gen556
2018-07-17_06-39-15pm_DS_TS_200008_DJ_1135_TR1.CSV.pqd	Yesterday at 18:42	39,1 MB	Gen556
2018-07-17_06-37-24pm_DS_TS_247511_VL_7005_Circ_1.CSV.pqd	Yesterday at 18:41	67 KB	Gen556
2018-07-17_06-05-18pm_DS_TS_247511_VL_7007_Circ_2.CSV.pqd	Yesterday at 18:07	462 KB	Gen556
2018-07-17_06-04-56pm_DS_TS_247511_VL_7007_TR1.CSV.pqd	Yesterday at 18:06	815 KB	Gen556
2018-07-17_05-55-44pm_DS_TS_247510_VL_7006_TR1.CSV.pqd	Yesterday at 17:56	932 KB	Gen556
2018-07-17_05-52-37pm_DS_TS_220004_4027-PTAB-389.CSV.pqd	Yesterday at 17:53	3,7 MB	Gen556
2018-07-17_07-44-01am_PTA_3_PERISOR_PL._1.CSV.pqd	Yesterday at 07:45	385 KB	Gen556
2018-07-17_02-46-04am_DS_TS__2_TUGLUI.CSV.pqd	Yesterday at 02:47	942 KB	Gen556
2018-07-17_02-31-47am_PTA_3_PERISOR_PL._1.CSV.pqd	Yesterday at 02:32	385 KB	Gen556
2018-07-17_01-56-32am_DS_TS__2_TUGLUI.CSV.pqd	Yesterday at 01:57	942 KB	Gen556
2018-07-17_01-23-26am_PTA_1_Scoala_Daneasa_001.CSV.pqd	Yesterday at 01:24	208 KB	Gen556
2018-07-17_01-02-05am_DS_TS__2_TUGLUI_30.06.2015.CSV.pqd	Yesterday at 01:03	537 KB	Gen556
2018-07-16_09-21-24pm_DS_TS__2_TUGLUI.CSV.pqd	16 Jul 2018 at 21:22	942 KB	Gen556

Figure 10. PQDM-WebApp – PQDIF files generated

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 10.

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

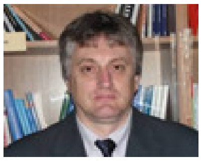
Claudiu Popirlan and Gabriel Stoian are members at Computer Science Department, University of Craiova and researchers at Laboratory of Formal Intelligence Integration in Analysis, Simulation, Development, Testing and Certification of Computation Infrastructures, INCESA (http://incesa.ro/#/computer_science/computation_infrastructure).

Claudiu Popirlan

Gabriel Stoian

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)

Leonardo Geo Manescu



Marian Ciontu

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

Miron Alba Adrian Cojoaca

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