DOI: 10.5604/01.3001.0010.8652

USING MICROSERVICES ARCHITECTURE AS ANALYTICAL SYSTEM FOR ELECTRICAL IMPEDANCE TOMOGRAPHY IMAGING

Tomasz Cieplak[1], Tomasz Rymarczyk[2,3], Grzegorz Kłosowski[1]
[1]Lublin University of Technology, Faculty of Management, Department of Organization of Enterprise
[2]Research and Development Center, Netrix S.A., Lublin, [3]University of Economics and Innovation in Lublin

Abstract. *An image reconstruction with use of EIT method has been found useful in many areas of medical, industrial and environmental applications. Papers show that computational systems used for image reconstructions are utilizing parallel and distributed computations and multi-tier architecture, as well as monolithic architecture. The aim of our research is to define an analytical system architecture that will be able to combine a variety of image reconstruction algorithms with their representations in different programming languages. Based on examples described in different proceedings and research papers, a microservices architecture seems to be an interesting alternative to the monolithic one.*

Keywords: electrical impedance tomography, microservices, Internet of Things, image reconstruction

ZASTOSOWANIE ARCHITEKTURY MIKROUSŁUG W ANALITYCZNYM SYSTEMIE REKONSTRUKCJI OBRAZÓW ELEKTRYCZNEJ TOMOGRAFII IMPEDANCYJNEJ

Streszczenie. *Zaprezentowano postępy prac związanych z budową system analitycznego służącego do rekonstrukcji obrazów obiektów badanych za pomocą elektrycznej tomografii impedancyjnej. Celem system jest elastyczność pozwalająca na integrację wewnątrz jednego system modułów analitycznych bazujących na różnych algorytmach rekonstrukcji obrazu identyfikowanego obiektu. Kolejnym ważnym wymaganiem jest możliwość oprogramowania modułów analitycznych za pomocą najczęściej wykorzystywanych w tej dziedzinie językach programowania. System zapewnia komunikację z urządzeniem za pośrednictwem łączy internetowych, co pozwala na zdalne sterowanie i pobieranie wyników pomiarów. Dodatkowym założeniem stawianym aplikacji jest możliwość korzystania z dowolnego źródła danych (urządzenie typu tomograf, baza danych, systemy plików) poddawanych analizie, a wyniki rekonstrukcji mają być dostępne dla każdego urządzenia komputerowego. Zaprezentowano dwa rodzaje architektury aplikacji, monolityczną i opartą o mikro usługi.*

Słowa kluczowe*: elektryczna tomografia impedancyjna, mikro usługi, Internet rzeczy, rekonstrukcja obrazów*

Introduction

Electrical Tomography is a relatively mature imaging method that can be useful in medical, industrial and environmental applications to discover the inner structure of an investigated object [11, 16, 23–27].

The image reconstruction in EIT is a highly ill-posed inverse problem [2, 7–9, 17–21]. To solve the EIT inverse task, several algorithms are well known and used. They can be split into two categories: direct algorithms and iterative ones. There are few recognized as ready to use for commercial applications (e.g. GREIT) [1] and many that are in continuous development, therefore risky to use. Every case of an image reconstruction consists of almost the same number of steps beginning from data acquisition, data analysis by means of different methods, and ending with a visualization. Using such assumption and combining information about similar researches carried out on analytical systems applications on different research fields [4, 12, 13], a thesis was formulated, that it is possible to use architecture based on microservices to combine various image reconstruction methods in one analytical system.

1. Monolithic approach to analytical system design

Till now, the natural way to create analytical system was to build a monolithic web application.

That design was used to create a prototype of system for tomography analysis. The basic assumption was made to build an application which will gather data from devices, preserve and analyze them, and accomplish basic visualizations. Additionally, application could monitor tomographic device status.

The system consisted of four layers, as shown in Figure 1. The data was acquired from tomographic devices as time series datasets. In this stage MQTT protocol was used to pass the data from the source to the analytical system.

The data flow (Figure 2) was replicated into two flows: the first one was sent to the database system (for this purpose NoSQL Cloudant database system was used) and the second one was sent to the analytical system for further analysis. The aim of such operation was to process data in real time as well as to keep raw historical data for additional offline analysis.

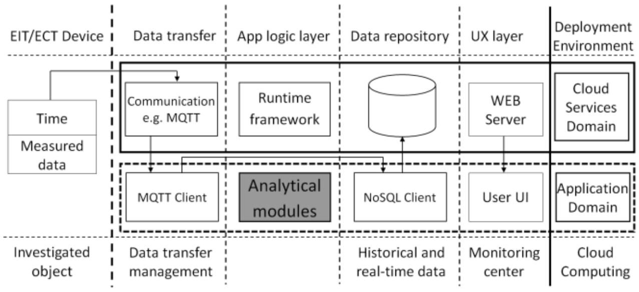


Fig. 1. Model of monolithic web application

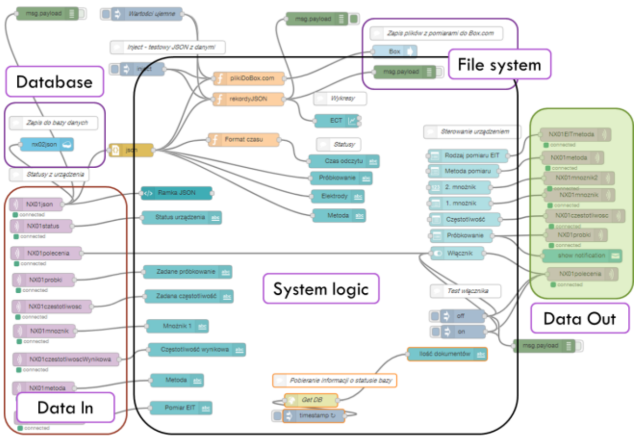
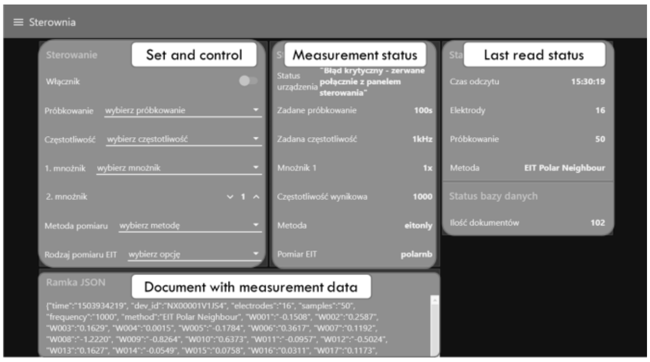


Fig. 2. Data flow inside of the system

The analytical system in form of web application (Figure 3) was built on top of IBM Bluemix PaaS platform. The advantage of this platform is easy integration with other analytical services like IBM Data Scientist Experience, which allows to use additional analytical methods from Apache Spark framework for further data analysis.

Monolithic application is a good approach, however, it had certain limitations. It was quite difficult to combine in one analytical stage few different algorithms written in different languages, which could be very useful for image reconstruction.



Sterowanie

Włącznik

Próbkowanie wybierz próbkowanie

Częstotliwość wybierz częstotliwość

1. mnożnik wybierz mnożnik

2. mnożnik 1

Metoda pomiaru wybierz metodę

Rodzaj pomiaru EIT wybierz opcję

Status urządzenia "Błąd krytyczny - zerwane połącznie z panelem sterowania"

Zadane próbkowanie 100s

Zadana częstotliwość 1kHz

Mnożnik 1 1x

Częstotliwość wynikowa 1000

Metoda eitonly

Pomiar EIT polarnb

Czas odczytu 15:30:19

Elektrody 16

Próbkowanie 50

Metoda EIT Polar Neighbour

Status bazy danych

Ilość dokumentów 102

Ramka JSON

{"time":"1503934219", "dev_id":"NX00001V1JS4", "electrodes":"16", "samples":"50",
"frequency":"1000", "method":"EIT Polar Neighbour", "W001":"-0.1508", "W002":"0.2587",
"W003":"0.1629", "W004":"0.0015", "W005":"-0.1784", "W006":"0.3617", "W007":"0.1192",
"W008":"-1.2220", "W009":"-0.8264", "W010":"0.6373", "W011":"-0.0957", "W012":"-0.5024",
"W013":"0.1627", "W014":"-0.0549", "W015":"0.0758", "W016":"0.0311", "W017":"0.1173",

Fig. 3. Web application – device's control page

The structure of such system also caused difficulties in extending analytical layer with additional methods, because it involved rewriting these methods in supported language or doing plumbing between system and additional methods libraries. The solution was scalable in means of adding additional application instances, memory, CPU etc. but it affected whole system due to monolithic nature, so there wasn't possible to scale only a part of the system (e.g. analytical resources). General problems with monolithic architecture:

- code becomes harder to maintain and test as it grows larger,
- coupling between modules causes random bugs when changes are made,
- deployments and upgrades require application downtime,
- if part of a service crashes then entire site goes down,
- difficulty in application scaling,
- incorporating new technologies is rather difficult.

2. Modular approach to analytical system design

The analytical system should easily deal with a variety of data sources, analytical methods, as well as many different types of users' applications. Taking into consideration the use cases and requirements of different analytical systems applications, the common features can be summarized as: need of easy deployment and monitoring, possibility of replacement of system components (if they are malfunctioned or a new version appears), reliability, flexibility and fault-tolerance. Experience brought from an industry (a good example is Netflix service) shows rapid change in applied, distributed systems architecture from building monolithic applications to adopting microservice architecture [5, 6, 14]. While the fundamental concepts of microservices are not new, sometimes they can be wrongly treated as SOA architecture, but in fact they are something different [5]. The aim of creating microservices concept was to eliminate unnecessary complexity and implement single functionality in one small and independent service. They can communicate with each other using mechanisms such as HTTP-based RESTful APIs. The adoption of microservices is driven by scalability challenges, lack of efficiency, slow development and difficulties with adoption of new technologies that arise in case of complex software systems, constructed as one huge monolithic application. After Martin Flower, it can be said that microservices are not only philosophy of designing systems but also a new way to cooperate with clients [10].

Microservices mean not only easiness and lack of problems caused by monolithic architecture. In case of heavily distributed system, developers need to follow strictly the design assumptions and keep standardization of the architecture. There are many patterns telling how the system should be built and decision of choosing one best for the designed system could be problematic. Adding to the bucket the variety of technology – languages, frameworks, deployment solutions – can cause the design process of a system based on microservices complicated [15]. On the other hand, the simplicity in the scaling of an application can repay a hard work. The deployment on microservices can be done easily with use of containerization technologies like a Docker [3]. Docker container contains everything required to run a software – code, runtime environment, system tools, libraries and settings. This ensures that the software will run the same way, regardless of where it's deployed. When the solution is divided into microservices and deployment procedure operates on containerization it is one step away from moving such system to a cloud computing environment [22].

3. The work aim

Nowadays, during a development stage of analytical systems, there are few requirements for the designed systems that should be taken into consideration:

- they should be reachable for every researcher,
- they are supposed to be scalable and easy extendable.

Analytical systems are utilized in every kind of business and industry. In case of the research on electrical tomography, one of the main parts of the system is an images reconstruction system. The measurement system and analytical system are separated in natural way. Also, modules of computational system can be divided into layers and analytical layer can be split into smaller parts including image's reconstruction methods. Each reconstruction method module should be easy to use and to replace.

Additionally, the system must be scalable in various ways. Need of scalability depends on a size of measurement system – microscale can consist of one measurement device but the same reconstruction system should be able to work with hundreds measurement sets. Such assumptions conduct one to look for the solution in a set of distributed systems architectures.

The summarization of system requirements gives a conclusion that all properties of building systems will be achieved with application of microservices architecture. Benefits of incorporating to the system the microservices architecture is as follows:

- if microservice crashes, the rest of the system will keep going,
- deployment doesn't require downtime of any part of the system,
- each microservice can be scaled individually according to its needs,
- services can use different technical stacks (i.e. programming language, runtime engine, operating system)
- code complexity is reduced.

This new approach is focused on analytical engine, as seen in Figure 4. Every single functionality is contained within one separated microservice – that is the premise and goal for the rebuilt system. In the case the system's rebuild one can assume several scenarios:

- adding new microservices when development is moving forward with application,
- breaking pieces of functionality into microservices over time,
- refactoring the entire system into microservices at once.

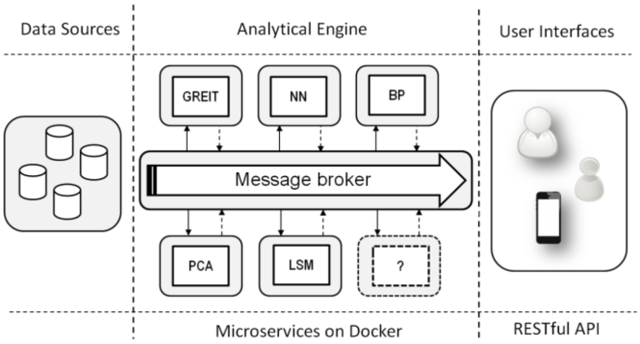


Fig. 4. System context diagram of the analytical system based on microservices

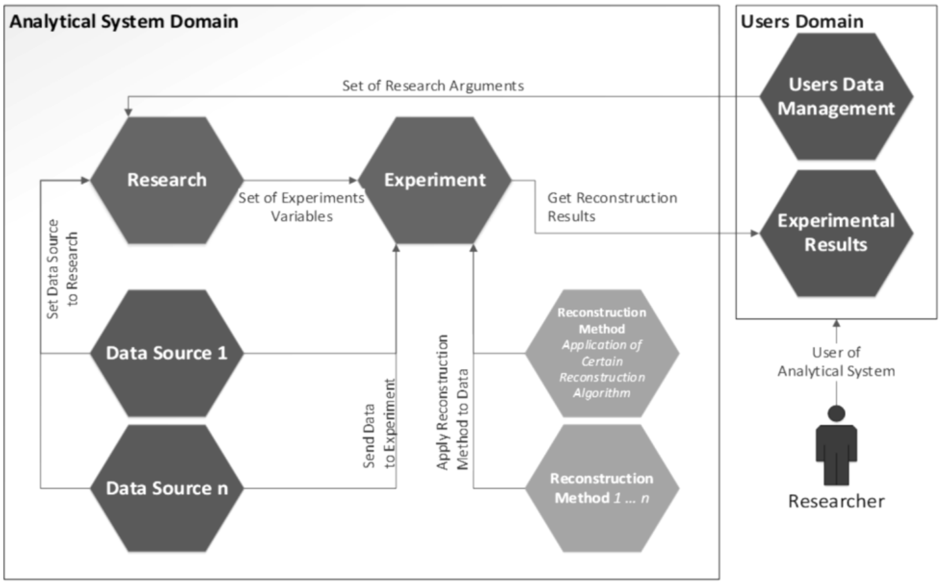
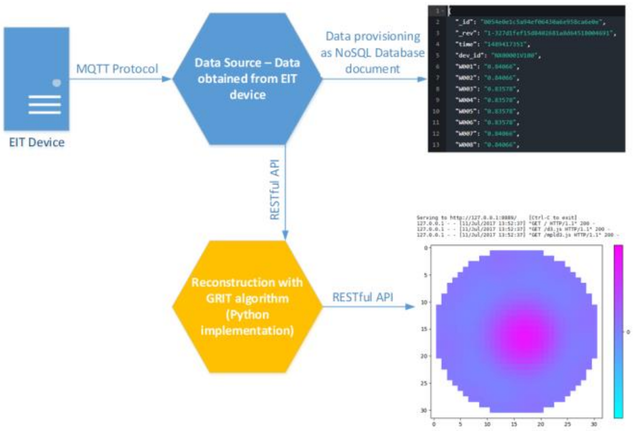

Analytical System Domain
Users Domain
Set of Research Arguments
Users Data Management
Research
Experiment
Set of Experiments Variables
Get Reconstruction Results
Experimental Results
Set Data Source to Research
Data Source 1
Data Source n
Send Data to Experiment
Apply Reconstruction Method to Data
Reconstruction Method Application of Certain Reconstruction Algorithm
Reconstruction Method 1 … n
User of Analytical System
Researcher

Fig. 5. Container diagram of analytical system


EIT Device
MQTT Protocol
Data Source – Data obtained from EIT device
Data provisioning as NoSQL Database document
RESTful API
Reconstruction with GRIT algorithm (Python implementation)
RESTful API
0
5
10
15
20
25
30
0
5
10
15
20
25
30
0

```
{
  "_id": "0054e0e1c5a94ef06430a6e958ca6e0e",
  "_rev": "1-327d1fef15d8402681a8d64518004691",
  "time": "1489417351",
  "dev_id": "0000001V100",
  "W001": "0.84066",
  "W002": "0.84066",
  "W003": "0.83578",
  "W004": "0.83578",
  "W005": "0.83578",
  "W006": "0.83578",
  "W007": "0.84066",
  "W008": "0.84066",
```

```
Serving to http://127.0.0.1:8889/    [Ctrl-C to exit]
127.0.0.1 - - [11/Jul/2017 13:52:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Jul/2017 13:52:37] "GET /d3.js HTTP/1.1" 200 -
127.0.0.1 - - [11/Jul/2017 13:52:37] "GET /mpld3.js HTTP/1.1" 200 -
```

Fig. 6. Component diagram of image reconstruction case with application of GREIT algorithm

While the first two approaches are feasible, the latter was not appropriate in case of the presented research.

Ultimately, microservices are communicating with each other and the rest of the system using message broker, which accept messages, queue them up and route them to destination(s). These analytical modules have they own APIs which expose calculation results to the system clients. In fact, the calculation result may be an image reconstruction matrix, which can be used by such libraries, like D3.js, to visualize a reconstruction image (in this case MPLD3 library was used). More detailed diagram of the system is shown in Figure 5. The core of the system is included in analytical system domain and user functionality is located inside users' domain.

Those two subsystems communicate with each other using RESTful API and data is exchanged in JSON format. To use the system, first, user must create a unit called Research. It is used as a container to store different researches carried by the same user. Each research can consist of many different experiments with different properties, like different source of data as well as variety of reconstruction methods. After setting research arguments and experiment properties user can run the experiment. As a result of the experiment, reconstruction of an image is ready for further processing. The final step is displaying the reconstruction image on the user's application on the device. User decides about the form in which the reconstruction data will be depicted (e.g. in web or mobile application) due to RESTful API. Modular system design allows to rebuild system easily, providing that the RESTful API and data format stays the same. Application of microservices permits using of polyglot programming, i.e. multiple programming languages inside one computational system.

Taking into consideration more detailed data flow, which is shown in Figure 6 (see page below), one can see some example of data analysis procedure. In the simplest case, data is gathered from EIT device and then stored in database (i.e. NoSQL database). Next, reconstruction algorithm consume data brought from data source and produce the reconstruction image. The image is presented on a web page. In this case only two microservices are used: one as data source and one for image reconstruction. The data source microservice is written in JavaScript using Node.js framework and the other is implemented in Python with Flask library.

4. Conclusions

The difference between monolithic and modular approaches to build an analytical system was shown. One comes to conclusion that monolithic application is good for research starting point, but quickly it becomes very restrictive in many areas.

An important factor in the use of architecture is the ability to verify the operation of different reconstruction algorithms running at the same time. To date, such experiences have been difficult.

An important feature of the analytical system based on microservices is its resistance to failures due to faulty analysis modules. In this case, failure of one analytical module does not cause the entire system to shutdown. What in the case of a monolithic system leads to the closure of the entire system.

This solution works with IBM Bluemix cloud computing system. By using the PaaS model, each microservice can receive virtually unlimited hardware resources. The number of running micro-instances and resources such as memory, processor type can be dynamically changed during system operation. In the case of a monolithic applications, such operations are difficult to carry out.

In the presented research, the main problem with a monolithic application was a lack of easy extensibility in adoption of analysis algorithms written in different programming languages. What, as shown, can be bypassed by changing the application architecture and use of microservices.

References

[1] Adler A., Arnold J.H., Bayford R., Borsic A., Brown B., Dixon P., Grychtol B.: GREIT: a unified approach to 2D linear EIT reconstruction of lung images. Physiological measurement 30(6), 2009, 35–55, [DOI: 10.1088/0967-3334/30/6/S03].

[2] Borcea L.: Electrical impedance tomography. Inverse Problems 18, 2002, 99–136.

[3] Amaral M., Polo J., Carrera D., Mohomed I., Unuvar M., Steinder M.: Performance evaluation of microservices architectures using containers. Network Computing and Applications (NCA), IEEE 14th International Symposium, 2015.

[4] D'Agostino D., Roverelli L., Zereik G., De Luca A., Salvaterra R., Belfiore A., Tiengo A.: A microservice-based portal for X-ray transient and variable sources. PeerJ Preprints, No. e2519v2, 2017.

[5] Dragoni N., Giallorenzo S., Lafuente A. L., Mazzara M., Montesi F., Mustafin R., Safina L.: Microservices: yesterday, today, and tomorrow. arXiv preprint arXiv:1606.04036, 2016.

[6] Dragoni N., Lanese I., Larsen S. T., Mazzara M., Mustafin R., Safina L.: Microservices: How to make your application scale. arXiv preprint arXiv:1702.07149, 2017

[7] Duda K., Adamkiewicz P., Rymarczyk T.: Nondestructive Method to Examine Brick Wall Dampness. International Interdisciplinary Phd Workshop 2016, 68–71.

[8] Filipowicz S.F., Rymarczyk T.: Measurement Methods and Image Reconstruction in Electrical Impedance Tomography. Przeglad Elektrotechniczny 88(6), 2012, 247–250.

[9] Filipowicz S.F., Rymarczyk T.: The Shape Reconstruction of Unknown Objects for Inverse Problems. Przeglad Elektrotechniczny 88(3A), 2012, 55–57.

[10] Fowler M.: Microservices. ThoughtWorks, http://martinfowler.com/articles/microservices.html, 2014 [06.09.2017].

[11] Holder D.S.: Electrical Impedance Tomography: Methods, History and Applications. Series in Medical Physics and Biomedical Engineering, London 2005.

[12] Johanson A., Flögel S., Dullo C., Hasselbring W.: OceanTEA: Exploring Ocean-Derived Climate Data Using Microservices. 6th International Workshop on Climate Informatics, National Center for Atmospheric Research in Bloulder, 2016.

[13] Kapusta P., Majchrowicz M., Sankowski D., Jackowska-Strumiłło L., Banasiak R.: Distributed multi-node, multi-GPU, heterogeneous system for 3D image reconstruction in Electrical Capacitance Tomography–network performance and application analysis. Przegląd Elektrotechniczny 89(2B), 2013, 339—342.

[14] Kim M., Mohindra A., Muthusamy V., Ranchal R., Salapura V., Slominski A., Khalaf R.: Building scalable, secure, multi-tenant cloud services on IBM Bluemix. IBM Journal of Research and Development 60(2-3), 2016.

[15] Richardson C.: Pattern: Microservices Architecture, Microservices.io. http://microservices.io/patterns/microservices.html [06.09.2017].

[16] Rybak G., Chaniecki Z., Grudzień K., Romanowski A., Sankowski D.: Non–invasive methods of industrial process control. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 3, 2014, 41–45 [DOI: 10.5604/20830157.1121349].

[17] Rymarczyk T.: Using electrical impedance tomography to monitoring flood banks. International Journal of Applied Electromagnetics and Mechanics 45, 2014, 489–494.

[18] Rymarczyk T.: New Methods to Determine Moisture Areas by Electrical Impedance Tomography. International Journal of Applied Electromagnetics and Mechanics 37(1–2), 2016, 79–87.

[19] Rymarczyk T., Tchórzewski P., Sikora J.: Monitoring of Flood Embankment System by Nondestructive Method with Infinite Boundary Element. Studies in Applied Electromagnetics and Mechanics 40, 2015, 176–183.

[20] Rymarczyk T., Tchórzewski P.: Topological methods to determine damages of flood embankments. Przegląd Elektrotechniczny 92(12), 2016, 153–156.

[21] Sankowski D., Sikora J.: Electrical capacitance tomography: Theoretical basis and applications. IEL, Warsaw 2010.

[22] Sousa G., Rudametkin W., Duchien L.: Automated Setup of Multi-Cloud Environments for Microservices-Based Applications. 9th IEEE International Conference on Cloud Computing, San Francisco, USA, 2016.

[23] Sikora J., Wójtowicz S.: Industrial and Biological Tomography: Theoretical Basis and Applications. IEL, Warsaw 2010.

[24] Smolik W.: Forward Problem Solver for Image Reconstruction by Nonlinear Optimization in Electrical Capacitance Tomography. Flow Measurement and Instrumentation 21, 2010, 70–77.

[25] Tai C., Chung E., Chan T.: Electrical impedance tomography using level set representation and total variational regularization. Journal of Computational Physics 205(1), 2005, 357–372.

[26] Wajman R., Fiderek P., Fidos H., Jaworski T., Nowakowski J., Sankowski D., Banasiak R.: Metrological evaluation of a 3D electrical capacitance tomography measurement system for two-phase flow fraction determination. Meas. Sci. Technol. 24(6), 2013, 065302.

[27] Wang M.: Industrial Tomography: Systems and Applications. Elsevier, 2015.

Ph.D. Tomasz Cieplak
e-mail: t.cieplak@pollub.pl

Doctor of Engineering, acting assistant professor in the Department of Enterprise Organization in the Faculty of Management, Lublin University of Technology. He has research experience gained in the prepared doctoral dissertation at Saga University, Japan. The area of research interest is the use model of cloud computing in the enterprise, creating systems based on SOA and the cloud computing model.

Ph.D. Eng. Tomasz Rymarczyk
e-mail: tomasz@rymarczyk.com

He is the director in Research and Development Center in Netrix S.A. He also is assistant professor in the University of Economics and Innovation, Lublin, Poland. His research area focuses on the application of non-invasive imaging techniques, electrical tomography, image reconstruction, numerical modelling, image processing and analysis, process tomography, software engineering, knowledge engineering, artificial intelligence and computer measurement systems.

Ph.D. Eng. Grzegorz Kłosowski
e-mail: g.klosowski@pollub.pl

Assistant Professor in Department of Organization of Enterprise at the Faculty of Management of Lublin University of Technology. The research interests of the author include artificial intelligence, simulation and modeling of engineering and business processes. Leader and participant in several implementation projects. Chairman of the Board of the Special Purpose Vehicle of POLLUB-Invest Ltd., whose purpose is indirect commercialization.

otrzymano/received: 10.10.2017 *przyjęto do druku/accepted: 15.02.2018*