

Applying process mining techniques and neural networks to creating and assessment of business process models

K Grigorova¹, K Mironov¹ and E Y Malysheva²

¹University of Ruse, Studentska street 8, Ruse, Bulgaria, 7017

²Volga Region State University of Services, Gagarina street 4, Togliatti, Russia, 445677

Abstract. The article presents an approach for automated generation of business process models by applying process mining techniques to event logs created during the operation of information systems used in an organization. Existing algorithms for process mining are discussed. Criteria for performing a comparative analysis of these algorithms are specified. A framework is proposed in which to build and analyze business process models. The framework includes tools for initial analysis of the event log file, extracting elements of a business process model, and composing a new model by applying a trained neural network.

1. Introduction

The rapid development of technology coupled with the proven benefits of implementing a process-oriented management approach are the causes of the growing use of business process models in organizational decision-making. Enhanced efficiency, better resource monitoring, and enhanced competitiveness are one of the reasons why business process management is increasingly involved into medium and small business organizations. Modern marketing methods, combined with software solutions that support and facilitate them, convey the understanding of business beyond the organization.

The need for businesses to get more information on how their processes are performed in a real environment is one of the main reasons for the spreading and use of process mining [1]. Modern business management systems and large volumes of data they generate create a favorable environment for the application of various process mining methods.

In addition to applying the most common algorithms and generating new business process models, analyzing system information allows to validate the integrity of the overall process, as well as to provide valuable metrics about its performance [2]. Another important feature of process mining is the ability to be used as a part of methodology for automating life cycle stages of business processes.

This article aims to present an experimental framework that uses process mining methods and enables automated generating and performance evaluating of identical business process models. A multilayer neural network is used for decision making during the business process generation as well as process evaluation according to the specific use case. In the previous related works the use of artificial neural networks is discussed as a tool for extracting business process models from event logs. The framework described in this paper involves neural network that is used on the stage of decomposing and evaluation of generated process model.

2. Process mining algorithms

The creation of a methodology for retrieving information from system data requires the knowledge of modification and application of the basic approaches of process mining [3]. Following is a review of the most commonly used algorithms for retrieving process models from system data, as well as analysis and classification based on their core characteristics and summarizing the most appropriate algorithms for specific use cases.

2.1. Alpha algorithm

The Alpha algorithm is one of the main in process mining [4]. It is characterized by creating a complete business process model by analyzing and processing event logs. The algorithm works by detecting Workflow nets and merging them into a complete logical structure. Typical for the algorithm is its fast performance, but it is also characterized by some limitations. The presence of a complete log file is required for the correctness of the generated business process model through Alpha miner. Incompleteness or lack of events, as well as describing of some exceptions in the event logs, may lead to incorrect final results. The Alpha algorithm is suitable for use in conjunction with systems that generate complete and well-structured event log files.

2.2. Heuristics miner

This algorithm is considered to be an extended version of the Alpha algorithm, but provides additional options for analyzing system event logs. The application of Heuristics miner is characterized by analyzing log data, sequencing events and determining the occurrence frequency. Based on these data, a network is created that consists of linked sequences and reflects the basic behavior of the workflow. Unlike the Alpha algorithm, Heuristics miner can be used for processing event logs with missing, incomplete and unassigned data, and even there are records of "emergency" situations.

2.3. Genetic miner

The genetic algorithm uses an evolutionary selection approach that mimics the process of natural evolution. This algorithm is significantly slower but offers optimal results based on system event log files [6]. It is characterized by analyzing and identifying initial "process candidates" in the form of related events, most often represented by Petri nets. Genetic miner has a cyclical character, and each of the potential processes is compared, and the one that comes closest to the general scenario in the event log is chosen. The genetic algorithm is suitable for use in system logs with missing or incomplete data, as well as with repeating events. It is not suitable for real-time application, and the final result is a detailed and very accurate business process model.

2.4. Fuzzy miner

Very often, business process models that are generated by event log analysis are large, unstructured and chaotic, without the ability to accentuate important events. The fuzzy algorithm is configurable and allows compact representation of detected patterns from different viewpoints (user or system) [7]. This is done through a set of techniques for identifying and removing isolated nodes, as well as merging several linked nodes into one. As a result of Fuzzy miner performance a logically true and identical copy of the original unstructured process is obtained, which is easy readable [5]. It is suitable for application on complete system log files without missing or incomplete information.

2.5. Selection of algorithm comparison criteria

When comparing algorithms for Process mining in the context of automated creation and assessment of business process models, consideration should be given to criteria that affect the program implementation as well as the number of characteristics of the input and output data. For this purpose, event logs and generated processes are classified according to the following criteria:

- Integrity - this characteristic indicates the ability of the algorithm to work in the case of missing event data.
- Uniqueness – indicates the ability of the algorithm to handle repeating events within a single process.

- Exceptions handling - indicates whether the algorithm can work normally when there are some exceptions.
- Real-time - indicates the ability of the algorithm to be used in real-time on a running system.
- Model integrity - indicates whether the generated model is a complete business process model or only a part of it (a series of related tasks).
- Abstraction level - indicates whether the obtained business process model is generalized and simplified.
- Structured model - indicates whether the obtained business process model is structured and suitable for reading by a human.

2.6. Comparative analysis of process mining algorithms

Automated generation and evaluation of business process models requires the processing event logs that differ by type and structure and the creation of a unified method for working –reading and analyzing the event log files. In order to determine the most appropriate algorithms for different use cases, process mining algorithms are compared and analyzed on criteria affecting their input and output data.

Using the specified criteria and taking into account the characteristics of process mining algorithms discussed in the previous sections, the results of the comparative analysis are summarized and presented in figure 1.

	Alpha Algorithm	Heuristics Miner	Genetic Miner	Fuzzy Miner
Integrity		✓	✓	
Uniqueness	✓	✓	✓	✓
Exceptions		✓	✓	
Real-Time	✓			
Model Integrity	✓	✓	✓	✓
Abstraction level				✓
Structured model				✓

Figure 1. Results from comparative analysis.

In the context of the framework created, when generating a business process model from an event log file, a preliminary analysis of the input data should be performed in order to determine which of the algorithms considered is the most appropriate for the specific case.

3. Framework for automated creation of business process model and evaluation of its performance

The automated creation of business process models combines the optimization and reengineering of existing models. Based on the fact that an activity can be implemented in several different ways, the purpose of the framework is to assemble different business process models that perform a particular business task. Components of each business process are already available business tasks extracted from system event log files. Each business process model will be evaluated for performance in order to identify the most effective for the considered business task.

Figure 2 shows a general architecture of the proposed framework. Each stage is considered and its main features are presented.

3.1. Pre-analysis

At this stage, a preliminary analysis of the system event log file to be processed is performed. The purpose is to determine its integrity and type. Based on the results of the preliminary analysis as well as those from the comparative analysis, an optimal process mining algorithm is selected to be applied in the specific case.

3.2. Extracting business process model

The selected process mining algorithm is applied to the event log file and a business process model is generated. In order to avoid future errors, the model is validated and its integrity is checked. In case

that a correct process is retrieved, it is simulated and evaluated for performance. These data will be used in the future to determine the most efficient process.

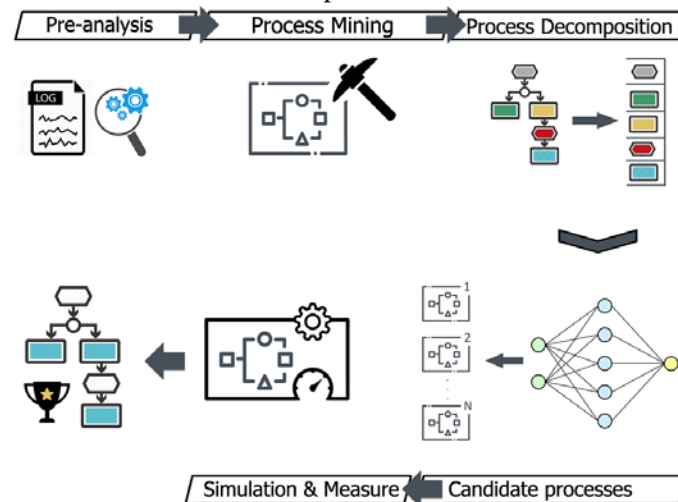


Figure 2. Framework's architecture.

3.3. Business process decomposition

The decomposition consists of segmenting the separate components of the model and storing them as independent elements. Intra-system storage of the elements allows the use of non relational model and, depending on the amount of business components, the application of Big data methods is recommended.

3.4. Generation of "candidate" processes

Creating processes that will be simulated and evaluated is performed using the decomposed business elements from the previous step and with the help of a neural network. Depending on the elements found in the decomposition of the model, the number of layers and neurons in the neural network may vary. The neural network training is based on the dependencies found in the log file, the relationships between the individual components of the process as well as the initial design and the subsequent redesign of the processes.

As a result of this phase, there are several logically identical "candidate" processes that perform the same functionality as the primary process generated by applying process mining algorithm.

3.5. Simulation and assessment

At this stage, it is intended to identify the most appropriate business process model for the particular business task. For this purpose, simulation is performed on each business process in a controlled and preconfigured work environment. During the execution of each process, its behavior is monitored and data is collected that will help to generate baseline assessment of performance.

3.6. Framework operation

After selecting a system event log file, which is the input data of the developed framework, a preliminary analysis is performed on it that aims to determine its structure, integrity, and availability of information about recurring events. After the analysis and on the basis of its results, a process mining algorithm is chosen which is most appropriate for the specific case. The selected algorithm is applied to the input log data and a business process model is derived. It is presented in XML-based format that allows easy processing. Additionally, metadata about the process logic is created used for initial neural network training.

The next step is decomposing the resulting business process model of components and storing them in a data repository, created as Redis database. Next the separate components are used to create "candidate" processes by using a neural network. The number of layers and neurons in the network is

determined dynamically depending on the case and the business process model under consideration. As an environment Java neural network framework Neuroph is used.

The generated "candidate" processes are simulated, taking into account their productivity metrics during the simulation [10]. After comparing the results obtained, the most effective business process model for the particular business task can be determined.

4. Creating an environment for automated generating and evaluating the performance of business process models

The presented framework for evaluating and generating business process models using Process Mining and Artificial Intelligence techniques hypothetically presents the possibilities for automated selection of the most effective business process models for a particular business task. The purpose of this experiment is to confirm the possibility of applying the developed framework, as well as to demonstrate in detail the preparation and implementation of the individual steps of the created algorithm.

4.1. Scope of the experiment

Creating an environment for automated generating and evaluating the performance of business process models covers the stages of decomposing, generating "candidate" processes, simulating and evaluating newly created models. Activities related to preliminary log file analysis and subsequent application of process mining algorithm are replaced by the use of real business processes. This ensures the applicability of the proposed framework in a real business environment.

For the purpose of the experiment, 4,000 business process models from BPMN (versions 1.1 and 2.0) and EPC were selected on a random basis. The processes are provided by the BPM Academic Initiative and are developed by various business and academic organizations.

4.2. Method of performance and expected results

Running the experiment follows the main steps of the proposed algorithm. Designed for automation, the algorithm allows many of the related activities to be automated through program implementation. A part of these processes are the validation, decomposition and storage of business process models. For the use of neural network decision-making, initial training and initialization of threshold values for individual neurons are envisaged. Simulating the generated business process models in a controlled environment allows tracking a number of their features that are taken into account when determining the most effective process for the particular task. Figure 3 shows the stages of the experiment, the basic software components used in each step, the input data used and the results obtained.

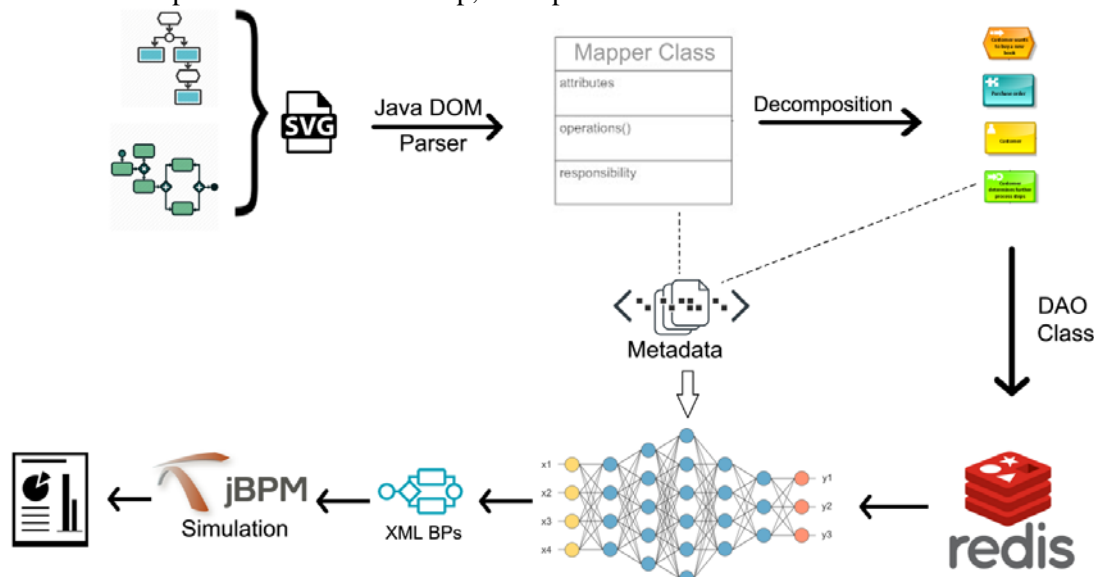


Figure 3. Experiment environment.

4.3. Pre-treatment and preparation of the BP for the experiment

The models used in the experiment need to be pre-treated to make them convenient. The initial state of the models in the database is in the form of vector files (SVG). The XML-based character of SVG files allows information to be extracted through a script using an XML-based software library. Within the experiment, this is done through the Java DOM Parser.

Intra-system presentation of information requires the creation of a Mapper class that allows the extracted data for each process to be represented as a Java object. The class contains a number of parameters, most commonly represented as arrays and collections of classes, designed to store the data specific to each model. Figure 4 shows the structure of the Mapper class.

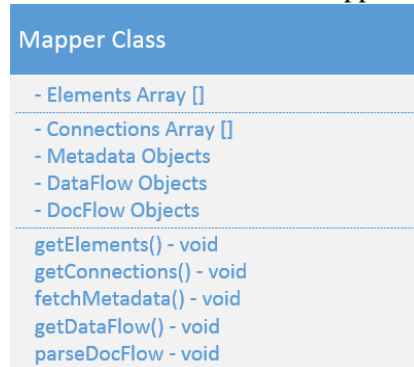


Figure 4. Mapper class structure.

As a result of the pre-processing of the models, an array that is composed of universal class instances is obtained. Each instance stores the transformed XML information for a business process. In this way, business data needed for the experiment is interpreted to an information array independent of file extensions and ready for the next step of the framework.

4.4. Decompiling of business processes

This stage of the developed framework aims to analyze each business process and segment and preserve its main elements. During the analysis, data about the model itself is derived. Later it serves as control results when compared to the generated "candidate" processes as well as is used to generate a data set used for primary training of the neural network. Before starting the decomposition process, a file containing basic metadata about the model is generated for each business process. Figure 5 shows the experimental structure of a JSON file storing metadata about a business process model.

The actual decomposition of the models is done by extracting all the data from the Mapper classes and presenting them as individual elements. Each element is analyzed and a meta model is generated that contains the data about it. Following is the classification and storing the element in a database. The generator automatically creates unique identifiers for each element and puts them in a "key-value" format, so the information is ready to be recorded into the database. The information presented in this way is submitted to the DAO classes that perform the actual storage of the decomposed items in a database. The experiment is conducted using the nonrelational Redis database, which allows fast storage and retrieval of the large volumes of business process models that are provided for the experiment.

4.5. Generation of "candidate" processes

Generating "candidate" processes involves building a logically identical business process to the original through the use of decomposed elements. For this purpose, metadata derived from business process analysis before decomposition, as well as the generated for each element, are used. From the complete meta model, the logic of the initial process is derived and the layers of the neural network are determined. Individual neurons are also determined on the basis of individual metadata for each element [11]. Within the experiment, the Java Neural Network framework - Neuroph is used for realization of this functionality. Figure 6 presents an example of a neural network and possible scenarios that individual neurons create.

```

1 {
2   "Standart": "BPMN",
3   "elements": {
4     "objects": [
5       {
6         "ID": 3,
7         "type": "Task",
8         "title": "Analyze order",
9         "metadata": [
10          {
11            "mId": 1,
12            "metadata": "status - pending"
13          }
14        ]
15      },
16      {
17        "ID": 4,
18        "type": "Parallel Gateway",
19        "title": "Concurrency",
20        "metadata": [
21          {}
22        ]
23      }
24    ],
25    "connections": [
26      {
27        "from": 3,
28        "to": 4,
29        "text": "",
30        "type": ""
31      }
32    ]
33  }
34 }

```

Figure 5. Metadata about business process model.

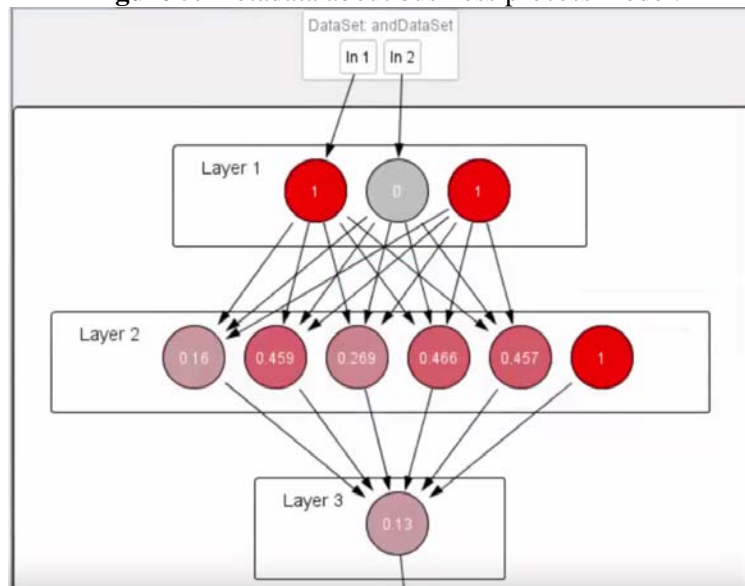


Figure 6. Neural network.

As a result of this stage, the generation of several business process models is expected, logically identical to the original BP model. Candidate processes are stored in an XML-based file that is used in process simulation.

4.6. Simulation and evaluation

This stage of the developed framework aims to determine the availability of more efficient models equivalent to the original one. This is done by simulating the original model and the generated "candidate" processes. The simulation is performed under the same conditions and the basic

parameters accompanying it are traced. For the purpose of the experiment, the Java-based Business Process Management framework - jBPM is used. The generated XML files of the "candidate" processes are loaded into the process server that will perform the simulation. Figure 7 and figure 8 show a view of the simulation and the resulting business process model within the jBPM framework.

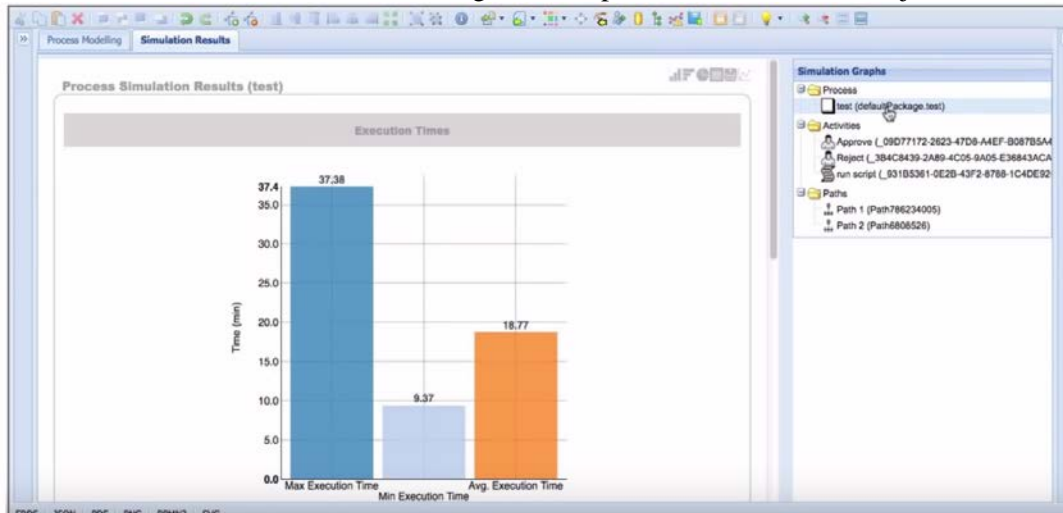


Figure 7. The results of business process simulation.

jBPM offers tracking of over 20 metrics related to process execution and generates a report of the results. By comparing the data about the initial process and those about the "candidate" processes, one can determine whether a more efficient model is found. Depending on the process requirements, performance parameters can be filtered and different performance criteria defined, according to the shortest implementation times, depending on the resources used, risk reduction, and more [8], [9].

The examined experiment shows the possibilities for practical application of the created framework for automated generation and evaluation of the efficiency of business process models. By combining the means of Process Mining and the capabilities of Artificial Intelligence, it is possible to create an environment for optimizing business processes and improving business performance.

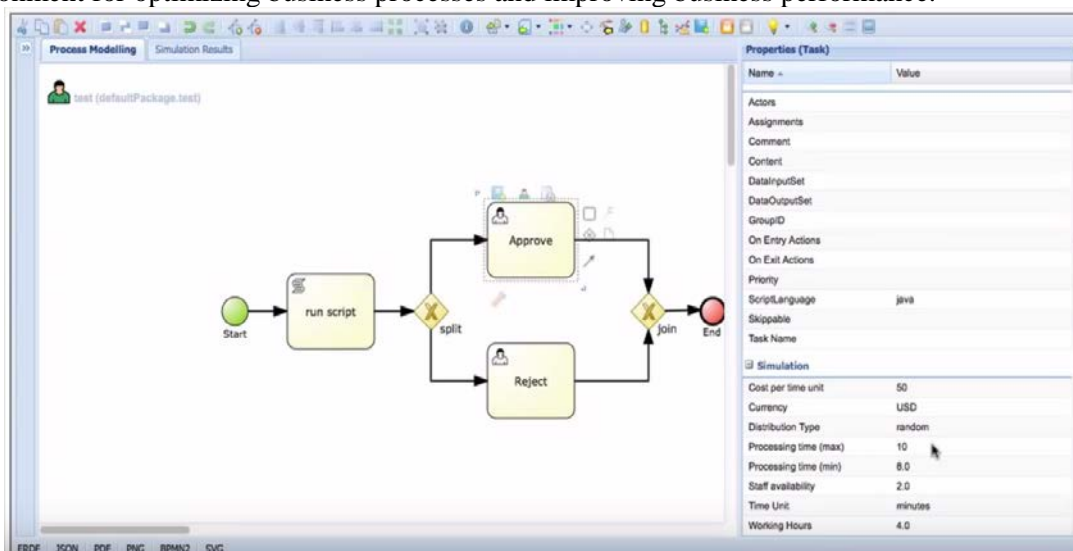


Figure 8. Resulting business process model.

5. Conclusion

The development of technology and increasing competitiveness in business environments contributes to the partial or complete automation of business tasks. The variety of business process modeling systems and the growing tendency for their use by business creates favorable conditions for the

application of process mining techniques. This article offers a solution for automated generating and evaluating the performance of business process models.

The presented framework uses process mining and artificial intelligence techniques and aims to automate part of the monitoring and optimization activities. As a final product of the examined framework, a business process model is generated that is the most effective for a particular business task. This increases the organization's competitiveness, and the automation reduces the resources used.

6. References

- [1] Wang J, Wong R K, Ding J, Guo Q and Wen L 2013 *IEEE Transactions on Services Computing* **6(4)** 484-496
- [2] van der Aalst W M, Reijers H A, Weijters A J, van Dongen B F, De Medeiros A A, Song M and Verbeek H M W 2007 *Information Systems* **32(5)** 713-732
- [3] Tiwari A, Turner C J and Majeed B 2008 *Business Process Management Journal* **14(1)** 5-22
- [4] van der Aalst W M, Weijters A J and Maruster L 2004 *IEEE Transactions on Knowledge and Data Engineering* **16(9)** 1128-1142
- [5] Weijters A J, van Der Aalst W M and De Medeiros A A 2006 *Technische Universiteit Eindhoven, Tech. Rep. WP* **166** 1-34
- [6] De Medeiros A A and Weijters A J 2005 *Lecture Notes in Computer Science* **3536**
- [7] van Dongen B F and Adriansyah A 2009 *Business Process Management Workshops* **43** 158-169
- [8] Maita A R C, Martins L C, López Paz C R, Peres S M and Fantinato M 2015 *Business Process Management Journal* **21(6)** 1391-1415
- [9] Hsiao B, Shu L, Young M and Yang H C 2016 *PACIS* p 264
- [10] Hornix P T 2007 *Performance analysis of business processes through process mining. Master Thesis, Eindhoven University of Technology*
- [11] Kopenkov V N and Myasnikov V V 2016 *Computer Optics* **40(5)** 713-720 DOI: 10.18287/2412-6179-2016-40-5-713-720

Acknowledgements

This work is supported by the National Scientific Research Fund under the contract DFNI - I02/13.