ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

INFORMATION TECHNOLOGY

DOI: 10.20998/2079-0023.2024.02.08
UDC 004.94

A. M. KOPP, Doctor of Philosophy (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Head of Software Engineering and Management Intelligent Technologies Department, Kharkiv, Ukraine; e mail: andrii.kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623
D. L. ORLOVSKYI, Candidate of Technical Sciences (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine; e mail: dmytro.orlovskyi@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-8261-2988
I. P. GAMAYUN, Doctor of Technical Sciences, Professor, National Technical University "Kharkiv Polytechnic Institute", Full Professor of Software Engineering and Management Intelligent Technologies Department, Kharkiv, Ukraine; e-mail: ihor.hamaiun@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-2099-4658
I. V. SAPOZHNYKOV, Student, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: illia.sapozhnykov@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0003-2802-1807

SOFTWARE DEVELOPMENT AND RESEARCH FOR MACHINE LEARNING-BASED STRUCTURAL ERRORS DETECTION IN BPMN MODELS

The most important tool for process management is business process modeling. Business process models allow to graphically represent the sequences of events, activities, and decision points that make up business processes. However, models that contain errors in depicting the business process structure can lead to misunderstanding of a business process, errors in its execution, and associated expenses. Thus, the aim of this study is to ensure the comprehensibility of business process models by detecting structural errors in business process models and their subsequent correction. During the analysis of the Business Process Management (BPM) lifecycle, it was found that the created business process models do not have a stage of control for the presence of errors in them. Therefore, the paper analyzes and improves the BPM lifecycle using the proposed approach. In the improved BPM lifecycle, it is proposed to take into account the correctness validation stage of business process models using the developed software. The paper proposes to process created BPMN (Business Process Model and Notation) models as connected directed graphs. To detect errors in business process models, one of the Machine Learning methods, K-Nearest Neighbors, is chosen, which is a fairly simple and effective classification method. The study also includes the software design and development, its performance validation, and usage to solve the given problem. To analyze the obtained results, the confusion matrix was used and the corresponding quality metrics were calculated. The obtained results confirm the suitability of the developed software for detecting structural errors in business process models. This web application, which is based on the created classification model, allows all interested users to upload business process models in BPMN 2.0 format, view the uploaded models, and analyze them for structural errors.

Keywords: business process models, structural errors, BPMN structural analysis, machine learning.

Introduction. Modern organizations are constantly improving their business processes to make them as efficient and cost-effective as possible. Business process modeling is a tool that businesses use to assess their current operations. In addition, business process modeling allows analysts to visualize a more efficient business process planned for implementation in the future. Business process modeling provides opportunities for improving the activities of the entire enterprise [1]. Business process modeling involves creating a visual representation of a business process. This is usually achieved with the help of business process modeling tools, such as flowcharts or specialized business process modeling notations, such as Business Process Model and Notation (BPMN) [1]. Business process modeling is used to identify improvements in business processes. This is done by depicting two different versions of a particular business process: the process as it exists now without changes (AS-IS); the process as it will be after improvements are made (TO-BE) [1].

Business process models containing structural errors may indicate errors in the actual processes they describe [1]. Timely detection and elimination of such errors will allow avoiding both financial losses and much more dangerous consequences of errors in critical business processes that may pose a threat to society, business, the state, and the environment. Thus, this study solves the relevant problem of identifying structural errors in business process models to ensure their quality and suitability for use.

The rest of this paper is organized as following: Section 2 includes literature analysis of recent studies, limitations, and approaches; Section 3 describes the object, subject, and considered research methods; Section 4 outlines the research results and analysis, Section 5 includes conclusions on the study.

Related work. Recent studies focused on business process model quality consider different structural metrics [2], application of fuzzy logic [3], clustering techniques based on complexity metrics [4], and verbal business process modeling guidelines extended by the various metrics and corresponding threshold values [5]. The IDEF0 methodology [6] is used to analyze the BPM lifecycle. The analysis of the "AS-IS" BPM lifecycle state starts with the IDEF0 context diagram (fig. 1), which defines inputs, outputs, controls (regulations, rules, policies, standards, etc.), and mechanisms (performers of activities and the required tools for this).

According to the presented context diagram (fig. 2), information about the business process is used as an input, and as a result of the business process execution, the corresponding indicator values are produced. The BPM lifecycle is shown in more detail in the decomposition diagram in fig. 2.

The demonstrated decomposition diagram (fig. 2) complements the inputs and outputs with a cyclic dependence – the obtained values of business process indicators are used to analyze its efficiency and, accordingly, create a new model of its future improved state. Moreover, fig. 2 shows the activity on creating business process models, business processes implementation and monitoring. The governance and mechanisms (e.g., business analyst, modeling notation, etc.) that regulate and support the relevant stages of the BPM lifecycle are defined (fig. 2).

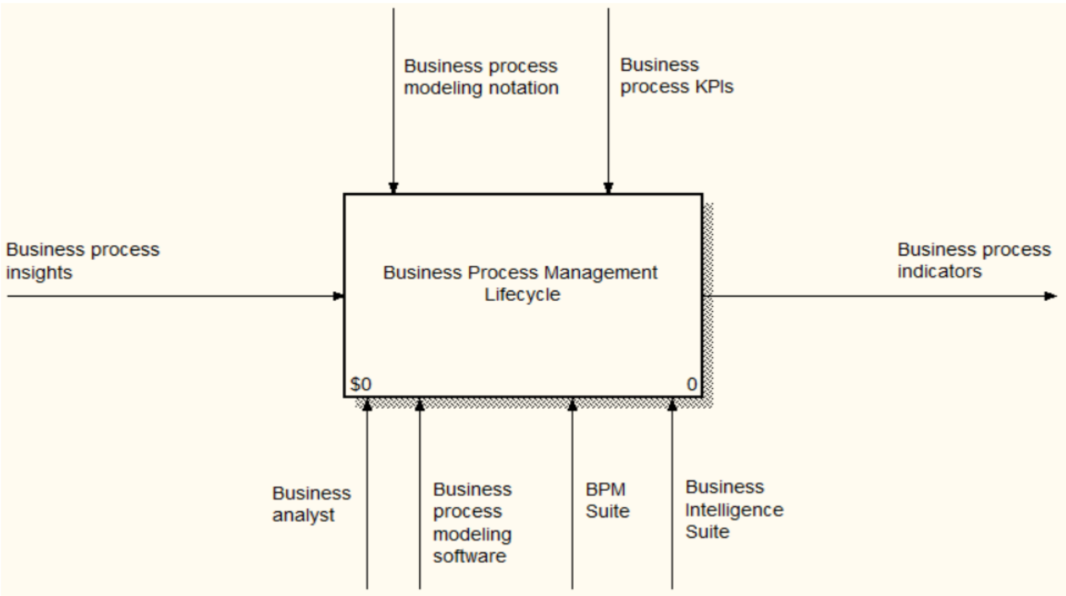


Fig. 1. The context diagram of the BPM lifecycle

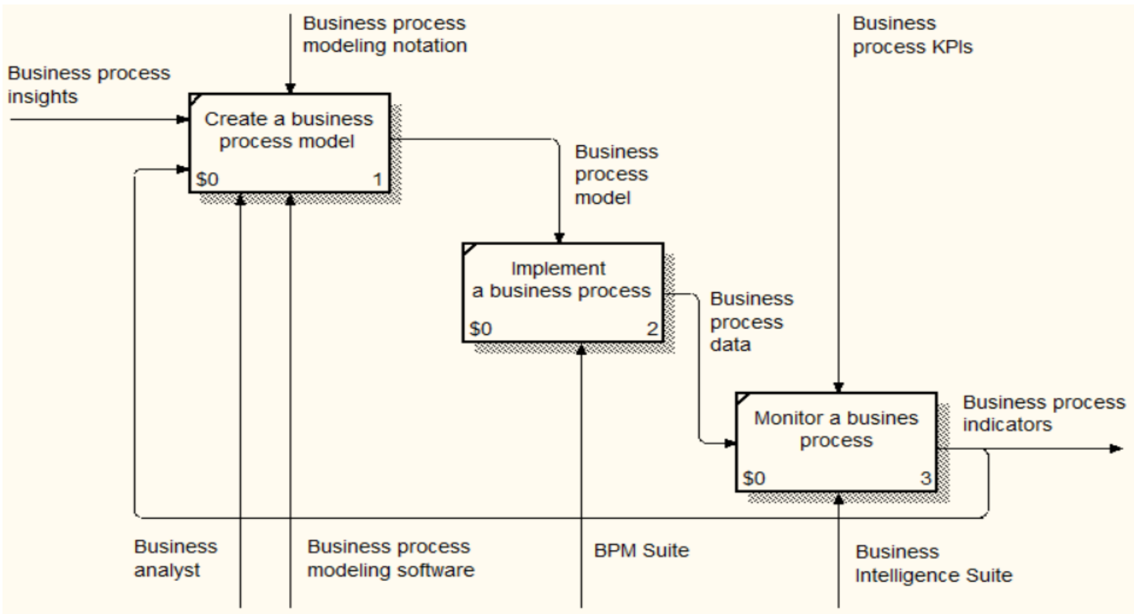


Fig. 2. The decomposition diagram of the BPM lifecycle

A detailed examination of the decomposition diagram (fig. 2) of the BPM lifecycle allows us to identify a "weakness" – the created models are immediately used to implement new business processes or planned changes in existing business processes without first checking for errors in these models (fig. 3).

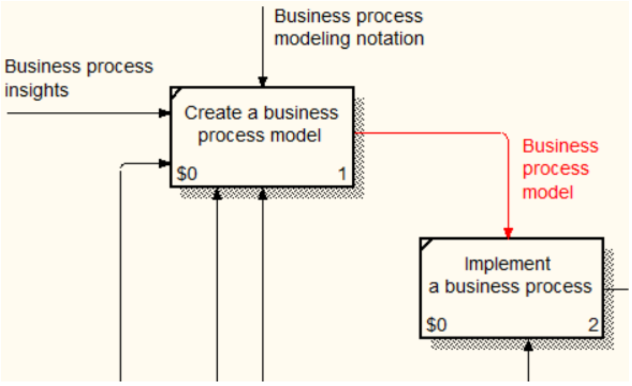


Fig. 3. The identified "weakness" of the BPM lifecycle

Thus, it is necessary to develop the algorithm and software to solve the problem of detecting structural errors in business process models.

Formally, a business process model given using the BPMN notation can be represented by a connected directed graph [7]:

$$BP = (N, A), \quad (1)$$

where:

N is the set of nodes – elements of a business process (events $E \subseteq N$, tasks $T \subseteq N$, gateways $G \subseteq N$, etc.);

A is the set of arcs – sequence flows of a business process that connect elements.

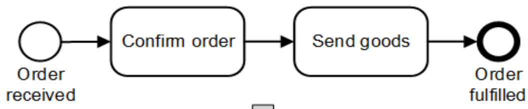
Since BPMN business process models are actually XML (eXtensible Markup Language) documents [8] created according to a special scheme, these models can be processed by computers relatively easily. Thus, the transformation of the BPMN model into a connected oriented graph (1) is possible by considering the "sequenceFlow" tags as graph arcs, and the "task" tags for actions, "event" for events, and "gateway" for gateways as graph nodes. A corresponding example of processing a BPMN file of the order processing business process model is shown in fig. 4.

High-quality business process models that do not contain errors must meet the following requirements [9]:

- use as few elements as possible in the business process model – if the business process model contains more than 50 elements, it should be decomposed into several simpler models;
- use one initial event;
- use one final event;
- do not use OR gateways.

Hence, formally, these rules of business process modeling can be represented as follows:

$$\theta = \begin{cases} 1, |N| > 50 \wedge |E^s| \neq 1 \wedge |E^e| \neq 1 \wedge |G^{or}| > 0, \\ 0, \text{else}. \end{cases} \quad (2)$$




```xml
<bpmn:process id="Process_0gge2c1" isExecutable="false">
  <bpmn:startEvent id="StartEvent_0qwsd25" name="Order received">
    <bpmn:outgoing>Flow_0bl2fsy</bpmn:outgoing>
  </bpmn:startEvent>
  <bpmn:task id="Activity_002ngbu" name="Confirm order">
    <bpmn:incoming>Flow_0bl2fsy</bpmn:incoming>
    <bpmn:outgoing>Flow_09orffj</bpmn:outgoing>
  </bpmn:task>
  <bpmn:sequenceFlow id="Flow_0bl2fsy" sourceRef="StartEvent_0qwsd25" targetRef="Activity_002ngbu" />
  <bpmn:task id="Activity_1yu2ggg" name="Send goods">
    <bpmn:incoming>Flow_09orffj</bpmn:incoming>
    <bpmn:outgoing>Flow_1y5ye81</bpmn:outgoing>
  </bpmn:task>
  <bpmn:sequenceFlow id="Flow_09orffj" sourceRef="Activity_002ngbu" targetRef="Activity_1yu2ggg" />
  <bpmn:endEvent id="Event_1g4nvnq" name="Order fulfilled">
    <bpmn:incoming>Flow_1y5ye81</bpmn:incoming>
  </bpmn:endEvent>
  <bpmn:sequenceFlow id="Flow_1y5ye81" sourceRef="Activity_1yu2ggg" targetRef="Event_1g4nvnq" />
</bpmn:process>
```

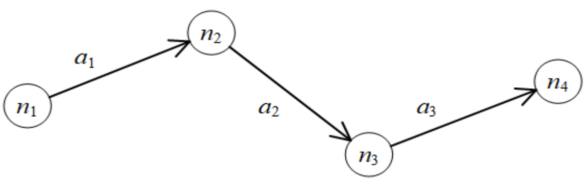


Fig. 4. The graph-based representation of the BPMN business process model

where:

- E^s is the set of start events of a business process, $E^s \subseteq E$;
- E^e is the set of end events of a business process, $E^e \subseteq E$;
- G^{or} is the set of "OR" logic gateways (inclusive) of a business process, $G^{or} \subseteq G$.

Therefore, if a business process model violates these rules (2), such a model can be classified as containing structural errors that make it difficult to understand and use by stakeholders. Furthermore, it can lead to errors in the business process itself, which is why these modeling errors must be detected and fixed.

Thus, each BPMN business process model from the training dataset can be assigned to one of the classes "has errors" ($\theta = 1$) or "has no errors" ($\theta = 0$) according to (2). It is proposed to detect structural errors in business process models using one of the well-known Machine Learning (ML) methods – K-nearest neighbors [10].

K-Nearest Neighbors (K-NN) is one of the simplest ML methods based on the supervised learning technique. The K-NN method determines the similarity between a new object and existing objects by adding the new object to the category that contains the most similar existing objects. The K-NN method stores all existing data and categorizes the new data object based on similarity. This means that when new data becomes available, it can be easily categorized using the K-NN method. The K-NN method can be used for both regression and classification, but it is mainly used for classification tasks. The K-NN method is based on a lazy learning method, because during the training phase it simply stores the dataset, and only when it receives new data for classification does K-NN assign it to the category that is most similar to the new data [10].

To apply the K-NN method to the problem of detecting structural errors in business process models, it is proposed to take as a basis the essential structural metrics of business process models:

- $|N|$ is the number of elements of a business process model (events, tasks, etc.);
- $|A|$ is the number of sequence flows that connect business process elements.

Hence, the distance between business process models is proposed to be measured using the following expression based on the Minkowski metric (3), which at $p = 2$ turns into the Euclidean distance, and standardization [11]:

$$d\left(BP^*, BP_i\right) = \left[\left|\left(\frac{|N^*| - \mu\left(|N_1|, \ldots, |N_m|\right)}{\sigma\left(|N_1|, \ldots, |N_m|\right)}\right) - \left(\frac{|N_i| - \mu\left(|N_1|, \ldots, |N_m|\right)}{\sigma\left(|N_1|, \ldots, |N_m|\right)}\right)\right|^p + \left|\left(\frac{|A^*| - \mu\left(|A_1|, \ldots, |A_m|\right)}{\sigma\left(|A_1|, \ldots, |A_m|\right)}\right) - \left(\frac{|A_i| - \mu\left(|A_1|, \ldots, |A_m|\right)}{\sigma\left(|A_1|, \ldots, |A_m|\right)}\right)\right|^p\right]^{\frac{1}{p}}, i = \overline{1, m}, \quad (3)$$

where:

- BP^* is the graph of the new BPMN model to be classified;
- BP_i is the graph of the i-th BPMN model from the training data set, $i = \overline{1, m}$;
- N^* is the set of elements of the new BPMN model;
- N_i the set of elements of the i-th model from the training set, $i = \overline{1, m}$;
- A^* is the set of sequence flows of the new BPMN model;
- A_i is the set of sequence flows of the i-th model from the training set, $i = \overline{1, m}$;
- μ is the arithmetic mean (4):

$$\mu(X) = \frac{1}{m} \sum_{i=1}^{m} x_i; \quad (4)$$

- σ is the standard deviation (5):

$$\sigma(X) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(x_i - \mu(X)\right)^2}; \quad (5)$$

- m is number of business process models in the training set.

The algorithm for detecting structural errors in business process models based on the K-NN method is shown in fig. 5 [10].

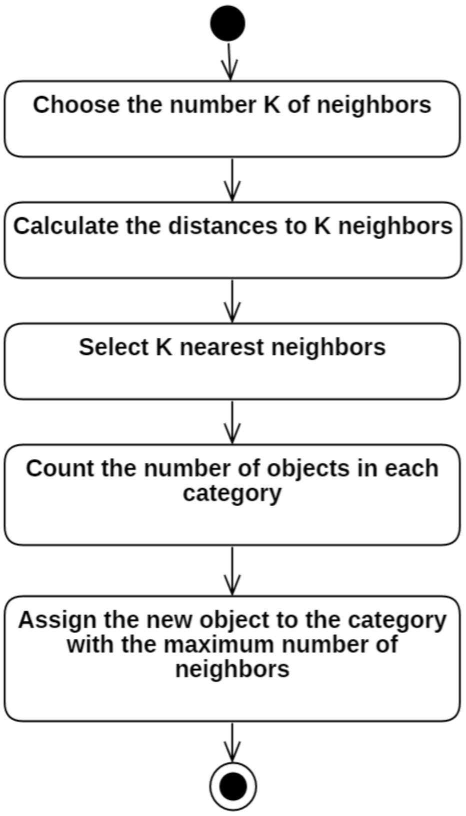


Fig. 5. The K-NN algorithm for detecting structural errors in business process models

Regarding the choice of the parameter, it is usually taken as $K = 5$ [10]. Thus, the algorithm based on the K-nearest neighbors method (fig. 5) is proposed to solve the problem of detecting structural errors in business process models and fixing them.

Materials and methods. This paper addresses a relevant problem of detecting structural errors in business process models to ensure their quality and suitability for use in organizational improvement or software development projects. Business process models that contain structural errors can cause difficulties in understanding them by stakeholders, which can result in the loss of all the benefits of modeling, analyzing, and improving of the real business processes.

The aim of this study is to ensure the comprehensibility of business process models by detecting structural errors in business process models and fixing them.

The research object is the process of detecting structural errors in business process models.

The research subject is the algorithm and software for detecting structural errors in business process models.

The decomposition diagram of the BPM lifecycle in the "TO-BE" state, which takes into account the need to control the quality of business process models and improve them using the proposed algorithm and software for solving the problem of detecting structural errors in business process models, is shown in fig. 6.

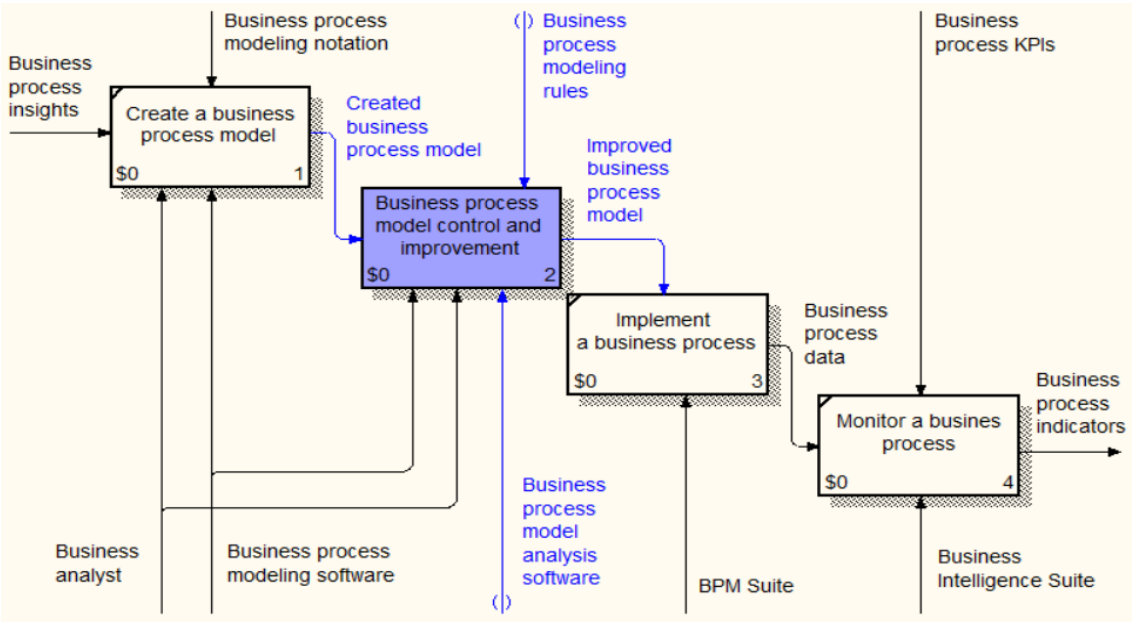


Fig. 6. The proposed improvement of the BPM lifecycle

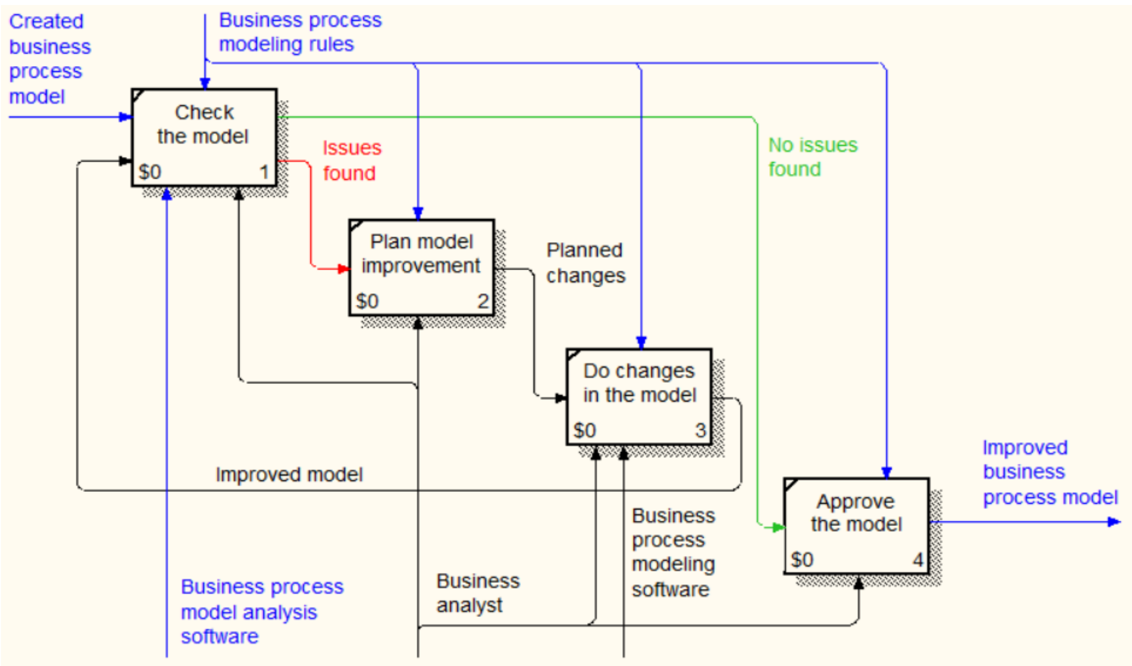


Fig. 7. The decomposition diagram of the business process model validation and improvement stage

The proposed changes (fig. 6) provide the intermediate stage to the BPM lifecycle, related to the validation of business process models and their improvement in case of structural errors. The decomposition diagram of the business process model validation and improvement stage (fig. 6) is shown in fig. 7.

Such validation is proposed to be carried out using a formal representation of BPMN models as connected directed graphs (1). The models from the training sample should be checked based on the rules (2) [9], and the rest of the business process models that will be checked by the software tool later, should be checked using the K-nearest neighbors method (fig. 5) to classify these models as having structural errors or not.

The proposed stage involves the following activities:

- analyze business process models using the developed software;
- plan changes in the business process model according to the identified errors;
- make changes to the business process model and re-analyze it;
- approve the business process model for further use if no errors were found during the last check.

users to allow them validate uploaded BPMN models and detect possible errors.

An activity diagram [12] demonstrating the process of user interaction with the software to detect structural errors in business process models is shown in fig. 9.

Let us demonstrate the results of using the created software “BPMN Assistant” to detect structural errors in business process models.

Results and discussion. The user starts working with the software by uploading a BPMN 2.0 model file that contains a description of the business process.

As an example, let us use the goods dispatch business process model, which is shown in fig. 10. This BPMN model was obtained from the public Camunda’s GitHub repository, which contains several thousands of business process models for research purposes [13].

The initial version of the considered business process model (fig. 10) shows that it contains certain structural flaws:

1) branching is used incorrectly (A, B);
2) the task is not connected to the process (C);
3) there are no end events, instead the workflow is completed by tasks (D, E, F).

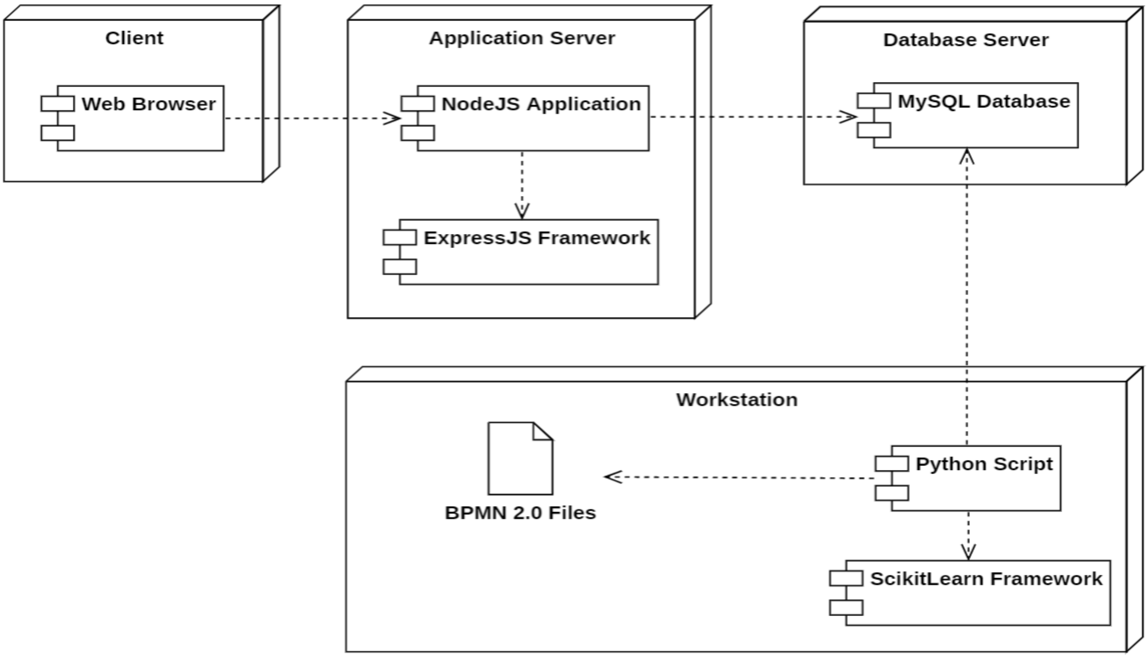


Fig. 8. The deployment diagram of the developed software components

The application “BPMN Assistant” is developed for end-users, responsible for BPMN models design and quality assurance. A deployment diagram of the software components [12], designed to validate business process models is shown in fig. 8.

As demonstrated in fig. 8, Python and Scikit Learn are used to detect structural errors in BPMN models using the K-NN algorithm and evaluate the obtained results, BPMN.IO is used to process BPMN 2.0 files, MySQL is used to store required data, Node JS and Express JS are used to build a client-server web application for business

The comparison of primary metrics of the initial and modified versions of the considered BPMN model is given in table 1.

Table 1 – The comparison of initial and modified BPMN model versions

BPMN model’s version	$\lvert N \rvert$	$\lvert E^s \rvert$	$\lvert E^e \rvert$	$\lvert G^{or} \rvert$	$\lvert A \rvert$
Initial	10	1	0	0	8
Modified	13	1	1	0	14

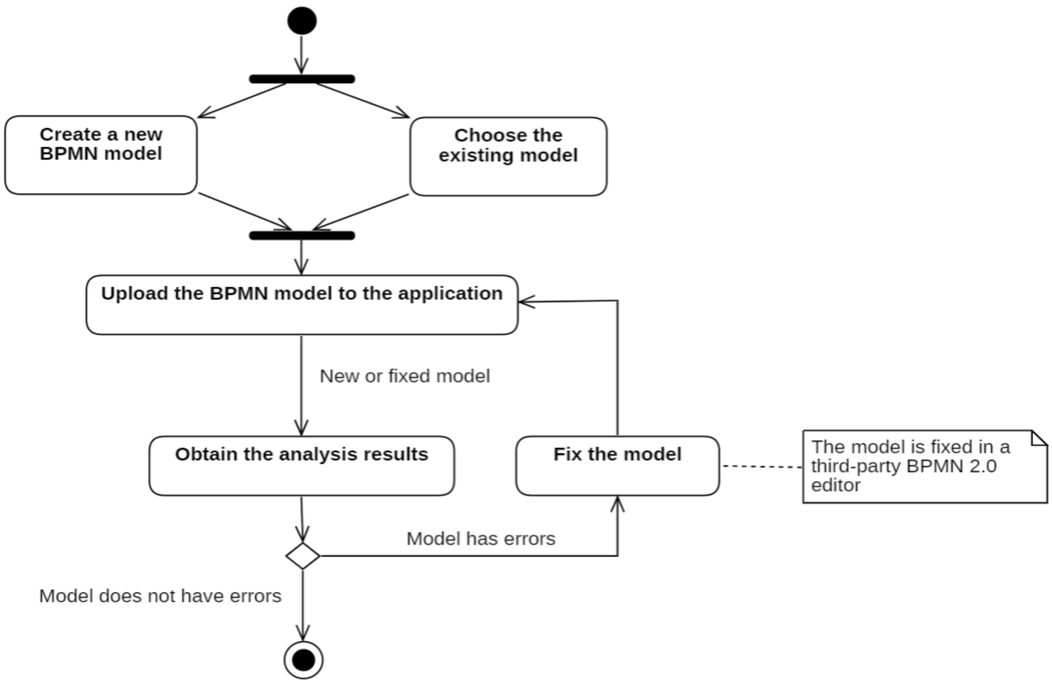


Fig. 9. The activity diagram of the user interaction with the developed software

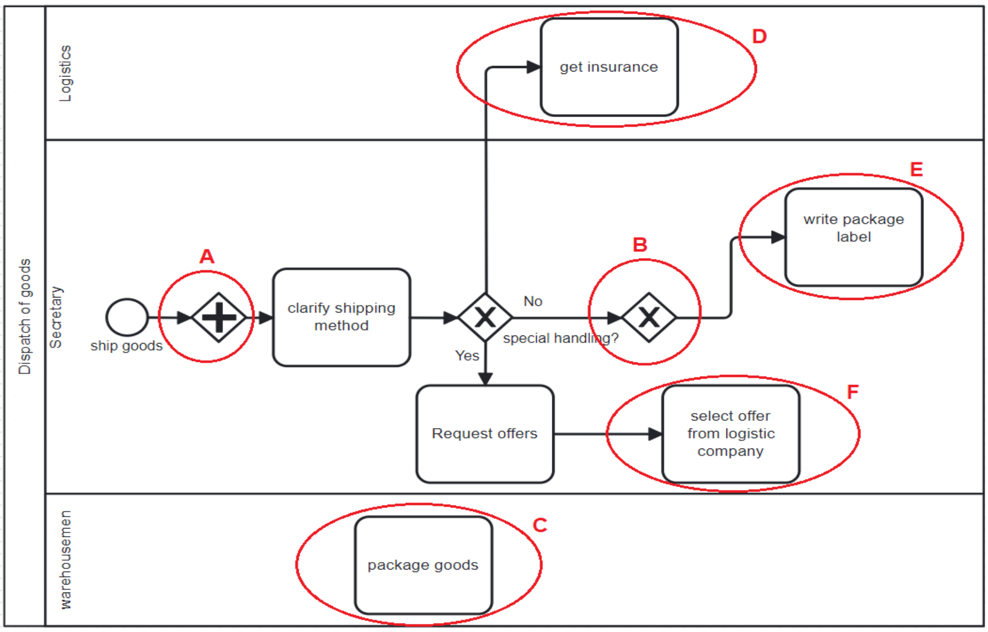


Fig. 10. The goods dispatch business process model with the detected structural errors

The modified version of the considered BPMN model is given in fig. 11.

To validate the performance of the developed software for detecting structural errors in business process models, we used the dataset of the Business Process Management Academic Initiative (BPMAI) project [14], which contains descriptions of 18812 BPMN models of various business processes.

Thus, using the data about business process models and their features, the possible presence of errors for each BPMN model was determined using formula (2). The original dataset was divided into 2 samples – training and test sets, which contain 75% (14109 models) and 25% (4703 models) of records respectively (fig. 12).

Further, using the Python programming language [15] and the Scikit Learn library [16], a model was built to detect structural errors in business process models based on the K-nearest neighbors classification method based on the BPMAI project training data set [14]. According to formula (3), the primary structural metrics of business process

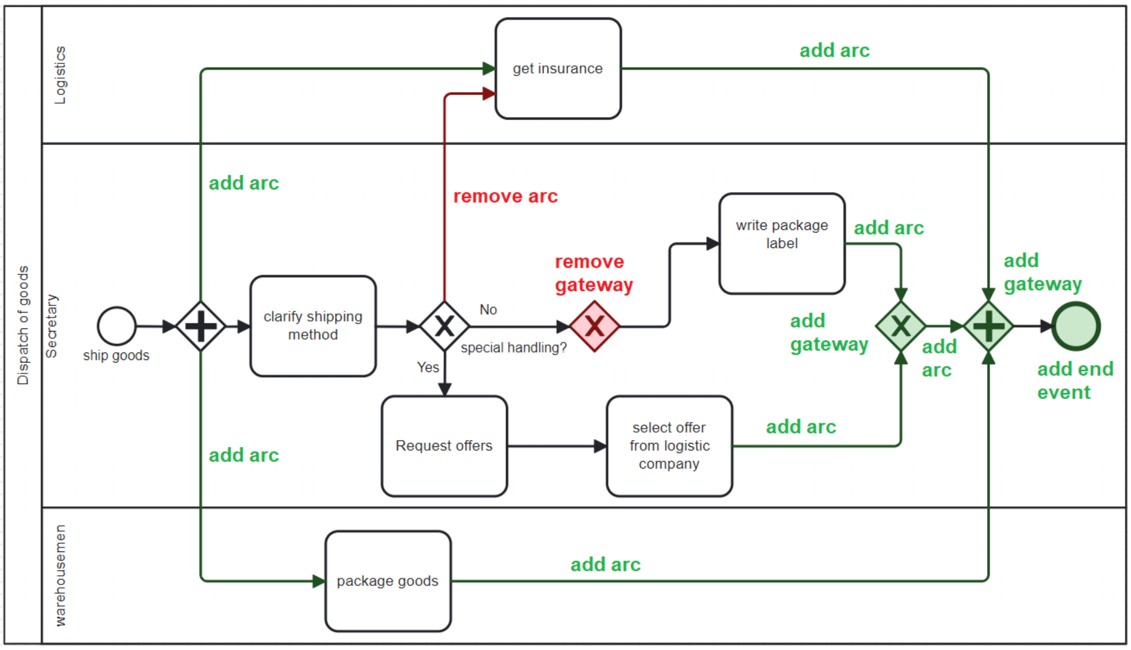


Fig. 11. The goods dispatch business process model with the modifications to eliminate structural errors

models used for classification are the number of elements $|N|$ and the number of sequence flows $|A|$.

The classification results are shown in fig. 13. The green color (category "1") represents BPMN models that have been identified as containing structural errors. To analyze the performance of the developed software for detecting structural errors in business process models, we will use the confusion matrix (table 2) [17].

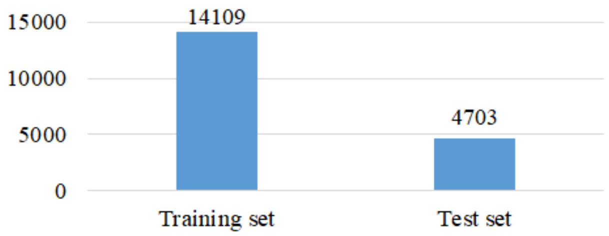


Fig. 12. The distribution of BPMN model data by training and test sets

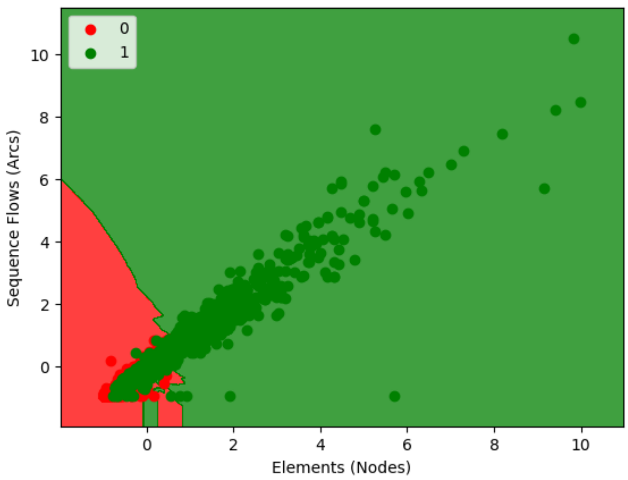


Fig. 13. The results of BPMN models classification using the K-NN method

Based on the obtained confusion matrix (table 2), the next measures were found [17]:

- true-positive (TP) results – 2716 models;
- false-positive (FP) results – 379 models;
- false negative (FN) results – 476 models;
- true-negative (TN) results – 1132 models.

Therefore, the following quality metrics [18] of the trained Machine Learning model were computed:

- precision (the share of correct answers within a class):

$$Precision = \frac{TP}{TP + FP} = 0.88; \quad (6)$$

- recall (the share of true-positive classifications):

$$Recall = \frac{TP}{TP + FN} = 0.85; \quad (7)$$

- accuracy (share of correct answers):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.82; \quad (8)$$

- summarized performance metric:

$$F - score = 2\frac{Precision \times Recall}{Precision + Recall} = 0.88. \quad (9)$$

Table 2 – The confusion matrix for K-NN performance analysis

Does a business process model have structural errors?	Correct answers	Yes	No
Classification results	Yes	2716	379
	No	476	1132

The obtained values of precision (4) and recall (5), as well as accuracy (6) and F-measure (7) are quite high, which makes it possible to consider the developed algorithm and software as suitable for use in detecting structural errors in business process models.

Conclusions and future work. This study addresses the relevant problem of ensuring the correctness of business process models by detecting structural errors in business process models. Thus, the following tasks were solved in the paper:

The stages of the BPM lifecycle are analyzed and the "weak spot" is identified – the created BPMN models tend to be immediately used to implement new business processes or planned changes in existing business processes without first checking for errors in these models

The algorithm for solving the problem of detecting structural errors in business process models is proposed – it is based on the K-NN Machine Learning method, taking into account primary BPMN model's size features: the number of elements (events, tasks, etc.) $|N|$ and the number of sequence flows (i.e. arcs) $|A|$.

The appropriate changes to the BPM lifecycle are proposed – it is proposed to be appended with the intermediate stage, provided to validate business process models and assume their improvement in case of structural errors detection.

The software architecture is designed and respective software components are developed using Python and Scikit Learn to use K-NN method, BPMN.IO to process BPMN 2.0 files, MySQL to manage data, and Node JS with Express JS to create the client-server web application for business users. The use of the developed software called "BPMN Assistant" to solve the problem of detecting structural errors in BPMN models is demonstrated – as a result of the performance validation of the developed algorithm and software, a sufficiently high quality of the K-NN classifier: $Precision = 0.88$ (6), $Recall = 0.85$ (7), $Accuracy = 0.82$ (8), and $F-score = 0.88$ (9) was obtained, which makes it possible to consider the proposed software solution as suitable for use in detecting structural errors in BPMN 2.0 business process models.

Future work assumes using other metrics and ML algorithms to achieve more accurate results of structural shortcomings detection in BPMN business process models.

References

1. Reijers H. A. *Business Process Management: The evolution of a discipline*. URL: https://doi.org/10.1016/j.compind.2021.103404 (access date: 17.09.2024).
2. Haj Ayech B., Ghannouchi S. A., El Hadj Amor E. A. *Extension of the BPM lifecycle to promote the maintainability of BPMN models*. URL: https://doi.org/10.1016/j.procs.2021.01.239 (access date: 17.09.2024).
3. Yahya F., Boukadi K., Ben-Abdallah H. *Improving the quality of Business Process Models: Lesson learned from the State of the Art*. URL: https://doi.org/10.1108/BPMJ-11-2017-0327 (access date: 17.09.2024).
4. Fotoglou C. et al. *Complexity clustering of BPMN models: initial experiments with the K-means algorithm*. URL: https://doi.org/10.1007/978-3-030-46224-6_5 (access date: 17.09.2024).
5. Corradini F. et al. *Correctness checking for BPMN collaborations with sub-processes*. URL: https://doi.org/10.1016/j.jss.2020.110594 (access date: 17.09.2024).
6. *IDEF0 Function Modeling*. URL: https://doi.org/10.1007/978-3-540-89556-5_5 (access date: 17.09.2024).
7. O'Regan G. *Graph Theory*. URL: https://doi.org/10.1007/978-3-031-26212-8_7 (access date: 17.09.2024).
8. *BPMN, Version 2.0*. URL: https://www.omg.org/spec/BPMN/2.0/PDF (access date: 17.09.2024).
9. Avila D. T., dos Santos R. I., Mendling J., Thom L. H. *A systematic literature review of process modeling guidelines and their empirical support*. URL: https://doi.org/10.1108/BPMJ-10-2019-0407 (access date: 17.09.2024).
10. *K-Nearest Neighbors*. URL: https://doi.org/10.1007/978-3-642-38652-7_2 (access date: 17.09.2024).
11. *Distance Measures*. URL: https://doi.org/10.1007/978-3-642-04898-2_626 (access date: 17.09.2024).
12. Unhelkar B. *Software Engineering with UML*. URL: https://doi.org/10.1201/9781351235181 (access date: 17.09.2024).
13. *BPMN for research*. URL: https://github.com/camunda/bpmn-for-research (access date: 17.09.2024).
14. *Model Collection of the Business Process Management Academic Initiative*. URL: https://doi.org/10.5281/zenodo.3758705 (access date: 17.09.2024).
15. *Python*. URL: https://www.python.org/ (access date: 17.09.2024).
16. *Scikit-learn*. URL: https://scikit-learn.org/ (access date: 17.09.2024).
17. *Confusion Matrix*. URL: https://doi.org/10.1007/978-0-387-30164-8_157 (access date: 17.09.2024).
18. Ting K. M. *Encyclopedia of Machine Learning*. URL: https://doi.org/10.1007/978-0-387-30164-8_652 (access date: 17.09.2024).

References (transliterated)

1. Reijers H. A. *Business Process Management: The evolution of a discipline*. Available at: https://doi.org/10.1016/j.compind.2021.103404 (accessed: 17.09.2024).
2. Haj Ayech B., Ghannouchi S. A., El Hadj Amor E. A. *Extension of the BPM lifecycle to promote the maintainability of BPMN models*. Available at: https://doi.org/10.1016/j.procs.2021.01.239 (accessed: 17.09.2024).
3. Yahya F., Boukadi K., Ben-Abdallah H. *Improving the quality of Business Process Models: Lesson learned from the State of the Art*. Available at: https://doi.org/10.1108/BPMJ-11-2017-0327 (accessed: 17.09.2024).
4. Fotoglou C. et al. *Complexity clustering of BPMN models: initial experiments with the K-means algorithm*. Available at: https://doi.org/10.1007/978-3-030-46224-6_5 (accessed: 17.09.2024).
5. Corradini F. et al. *Correctness checking for BPMN collaborations with sub-processes*. Available at: https://doi.org/10.1016/j.jss.2020.110594 (accessed: 17.09.2024).
6. *IDEF0 Function Modeling*. Available at: https://doi.org/10.1007/978-3-540-89556-5_5 (accessed: 17.09.2024).
7. O'Regan G. *Graph Theory*. Available at: https://doi.org/10.1007/978-3-031-26212-8_7 (accessed: 17.09.2024).
8. *BPMN, Version 2.0*. Available at: https://www.omg.org/spec/BPMN/2.0/PDF (accessed: 17.09.2024).
9. Avila D. T., dos Santos R. I., Mendling J., Thom L. H. *A systematic literature review of process modeling guidelines and their empirical support*. Available at: https://doi.org/10.1108/BPMJ-10-2019-0407 (accessed: 17.09.2024).
10. *K-Nearest Neighbors*. Available at: https://doi.org/10.1007/978-3-642-38652-7_2 (accessed: 17.09.2024).
11. *Distance Measures*. Available at: https://doi.org/10.1007/978-3-642-04898-2_626 (accessed: 17.09.2024).
12. Unhelkar B. *Software Engineering with UML*. Available at: https://doi.org/10.1201/9781351235181 (accessed: 17.09.2024).
13. *BPMN for research*. Available at: https://github.com/camunda/bpmn-for-research (accessed: 17.09.2024).
14. *Model Collection of the Business Process Management Academic Initiative*. Available at: https://doi.org/10.5281/zenodo.3758705 (accessed: 17.09.2024).

15. *Python*. Available at: https://www.python.org/ (accessed: 17.09.2024).
16. *Scikit-learn*. Available at: https://scikit-learn.org/ (accessed: 17.09.2024).
17. *Confusion Matrix*. Available at: https://doi.org/10.1007/978-0-387-30164-8_157 (accessed: 17.09.2024).
18. Ting K. M. *Encyclopedia of Machine Learning*. Available at: https://doi.org/10.1007/978-0-387-30164-8_652 (accessed: 17.09.2024).

Received 15.11.2024

УДК 004.94

А. М. КОПП, доктор філософії (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: andrii.kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623

Д. Л. ОРЛОВСЬКИЙ, кандидат технічних наук (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», професор кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: dmytro.orlovskyi@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-8261-2988

І. П. ГАМАЮН, доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», професор кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: ihor.hamaiun@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-2099-4658

І. В. САПОЖНИКОВ, Національний технічний університет «Харківський політехнічний інститут», студент, м. Харків, Україна; e-mail: illia.sapozhnykov@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0003-2802-1807

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ СТРУКТУРНИХ ПОМИЛОК У BPMN-МОДЕЛЯХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Найважливішим інструментом управління процесами є моделювання бізнес-процесів. Моделі бізнес-процесів дозволяють графічно представляти послідовності подій, дій і точок прийняття рішень, з яких складаються бізнес-процеси. Однак моделі, які містять помилки в представленні структури бізнес-процесу, можуть призвести до неправильного розуміння бізнес-процесу, помилок у його виконанні та пов’язаних з цим витрат. Таким чином, метою даного дослідження є забезпечення зрозумілості моделей бізнес-процесів шляхом виявлення структурних помилок у моделях бізнес-процесів та їх подальшого виправлення. Під час аналізу життєвого циклу управління бізнес-процесами (Business Process Management, BPM) виявлено, що для створених моделей бізнес-процесів не передбачено етапу контролю на наявність у них помилок. Таким чином, в статті аналізується та вдосконалюється життєвий цикл BPM, використовуючи запропонований підхід. У вдосконаленому життєвому циклі BPM пропонується врахувати етап перевірки коректності моделей бізнес-процесів за допомогою розробленого програмного забезпечення. У статті пропонується обробляти створені моделі BPMN (Business Process Model and Notation) у вигляді зв’язаних орієнтованих графів. Для виявлення помилок у моделях бізнес-процесів вибирається один із методів машинного навчання K-Nearest Neighbors, який є достатньо простим і ефективним методом класифікації. Дослідження також включає проектування та розробку програмного забезпечення, перевірку його продуктивності та застосування для розв’язання поставленої задачі. Для аналізу отриманих результатів було використано матрицю помилок та розраховано відповідні метрики якості. Отримані результати підтверджують придатність розробленого програмного забезпечення до виявлення структурних помилок у моделях бізнес-процесів. Веб-застосунок, який базується на побудованій моделі класифікації, дозволяє всім зацікавленим користувачам завантажувати моделі бізнес-процесів у форматі BPMN 2.0, переглядати завантажені моделі та аналізувати їх на наявність структурних помилок.

Ключові слова: моделі бізнес-процесів, структурні помилки, структурний аналіз BPMN, машинне навчання.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Копп Андрій Михайлович / Kopp Andrii Mykhailovych
Автор 2 / Author 2: Орловський Дмитро Леонідович / Orlovskyi Dmytro Leonidovych
Автор 3 / Author 3: Гамаюн Ігор Петрович / Gamayun Igor Petrovych
Автор 4 / Author 4: Сапожников Ілля Віталійович / Sapozhnykov Illia Vitaliiovych