Towards algorithmic and software solution for business process model analysis and correction

Olha Yanholenko†, Andrii Kopp*,†, Dmytro Orlovskyi† and Uliya Litvinova†

National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61002, Ukraine

Abstract

The Business Process Management (BPM) market is expected to grow by one-fifth over the next ten years due to the ongoing trend toward digitization. As a result, the number of critical BPM issues is also growing, including business process modeling and the problem of proper business process model design. In general, business process modeling is the key BPM technique used to capture, visualize, communicate, and analyze organizational workflows. With high-quality business process models, organizations will avoid problems with the understandability or maintainability of their workflows and related software applications. Therefore, this study aims to contribute to the BPM domain by proposing an algorithmic and software solution for analyzing and correcting business process models. This solution takes into account structural properties of Business Process Model and Notation (BPMN) diagrams, represented as graphs, evaluated against modeling rules and metrics, and modified to correct detected inefficiencies. The developed software tool is used to evaluate the performance of the proposed approach. The results are analyzed, conclusions are drawn, and future work is formulated.

Keywords

Business Process Model, Structural Analysis, Process Model Correction, Software Solution

1. Introduction

Business Process Management (BPM) is a key approach to organizational management that plays a very important role in operational activities and business performance improvement. Business process modeling is the fundamental technique of BPM used to visualize, analyze, and improve organizational activities. Throughout the history of BPM, several notations, standards, and methodologies have been proposed, such as IDEF, DFD (Data Flow Diagrams), EPC (Event-driven Process Chains), and others. However, the Business Process Model and Notation (BPMN) has been the most widely adopted and has become the de facto standard in business process modeling due to its rich set of standardized elements and its ability to clearly represent complex process interactions [1].

MoDaST-2024: 6th International Workshop on Modern Data Science Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

* Corresponding author.
† These authors contributed equally.
olha.yanholenko@khpi.edu.ua (O. Yanholenko); andrii.kopp@khpi.edu.ua (A. Kopp); dmytro.orlovskyi@khpi.edu.ua (D. Orlovskyi); uliya.litvinova@khpi.edu.ua (U. Litvinova)
0000-0001-7755-1255 (O. Yanholenko); 0000-0002-3189-5623 (A. Kopp); 0000-0002-8261-2988 (D. Orlovskyi); 0000-0001-6680-662X (U. Litvinova)

Quality assurance and correctness of designed BPMN models is extremely important to align business and information technology aspects, detect inefficiencies in workflows, train new employees, guide process improvements [2]. Using quality analysis techniques, it is possible to detect errors in created BPMN models, improve workflow scenarios, and improve the overall organizational performance. Therefore, a wide range of studies is devoted to the development of methods and tools for BPMN model quality evaluation, complexity assessment, correctness analysis and improvement, policy compliance analysis, and other quality aspects of business process models that may affect the efficiency of BPM practices [3].

The object of this study is the process of analysis and correction of BPMN diagrams. The subject of this study is the algorithmic and software solution for analysis and correction of business process models. This study aims to contribute to BPM practices by proposing the tool for improving the correctness of business process models based on their structural analysis.

Section 2 outlines the state of the art, including related work in the area of business process modeling and quality analysis of BPMN models, as well as the motivation and formal problem statement; Section 3 describes the algorithmic foundations of the proposed software solution, including the formal business process definition, modeling rules and corresponding metrics, and the process graph modification algorithm; Section 4 presents the results and discussion, including the dataset description, performance analysis, efficiency evaluation of the proposed approach, and examples of its application to real BPMN diagrams; finally, Section 5 presents conclusions were made and formulated future work directions.

2. State-of-the-Art Review

2.1. Related W

Business process modeling technique is used to visually describe and examine workflow tasks and sub-processes (i.e. activities), collaboration between separate business processes, and high-level process landscapes within organizations and enterprises. The main purpose of business process models is to capture, understand and improve organizational activities in order to eliminate possible problems, improve efficiency and effectiveness of business processes [3].

According to [4], BPMN is the most widely used business process modeling notation and has been widely adopted in industry and academia due to its powerful capabilities for graphical design and comprehensible description of complex workflows. BPMN provides a common visual notation (or language, according to some sources) for both business analysts and information technology professionals, bridging the communication gap between these parties and allowing them to collaborate on business process design and improvement activities [4].

The workflow design principles of BPMN are based on such elements (Fig. 1), as [5]:

- Activities (Tasks and Sub-Processes);
- Gateways (i.e. logical connectors, inclusive – OR, exclusive – XOR, parallel – AND);

- Events (Start, Intermediate, End);
- Flows (Sequence Flows, Message Flows);
- Pools (to separate workflows);
- Lanes (to separate responsibility areas);
- Data Objects and connecting Associations.

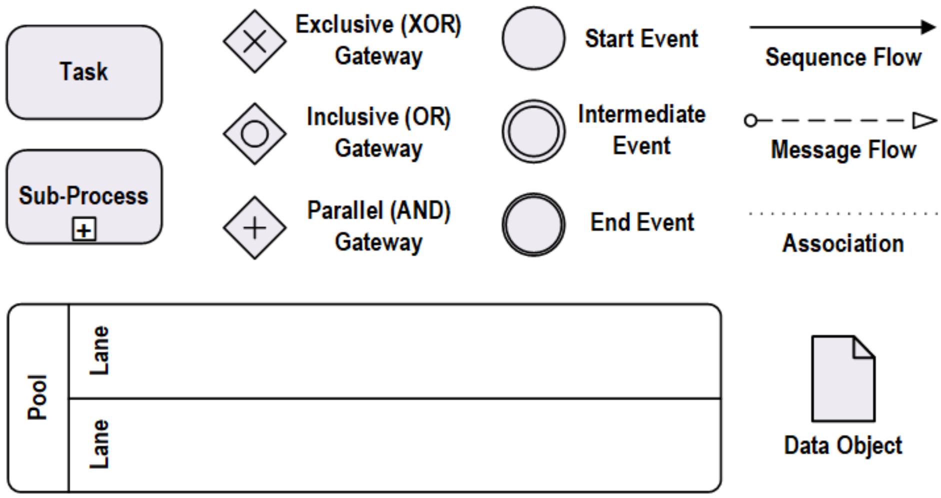

Figure 1: Key BPMN modeling elements.

These elements (Fig.1) serve to clearly describe the workflow, including activities, decision points (based on gateways), and collaborations, so that all stakeholders can clearly understand the logic of a business process and its behavior in different scenarios [5]. Using the rich set of modeling primitives, business analysts can address all business process modeling needs: tasks or sub-processes to represent workflow activities, gateways to model decision points, events to initiate or terminate business process executions, and flows to represent sequential scenarios and information exchanges within an organization [6].

Recent research has made significant contributions to the fields of BPM and business process modeling, especially in the areas of business process model complexity assessment, correctness analysis and verification, quality evaluation based on modeling guidelines, and BPMN model redesign. In [7], Fotoglou et al. proposed a method to assess the complexity of business process models (especially BPMN models) based on cluster analysis techniques (K-means algorithms) and three BPMN complexity metrics [7]:

- NOAJS (Number of Activities, Joins, and Splits);
- CFC (Control Flow Complexity);
- CNC (Coefficient of Network Connectivity).

Other researchers Kbaier and Ghannouchi in their paper [8] analyzed the quality of BPMN business process models using data mining techniques. Their study focuses on determining thresholds for BPMN quality metrics that will help modelers to evaluate the created business process models [8]. The most interesting finding is that thresholds vary in different domains, so the authors of [8] also used clustering techniques to determine domain-specific thresholds for BPMN quality metrics. Corradini et al [9] propose research on BPMN models with sub-process correctness checking. In their study [9], the authors focus on the problem of providing the BPMN correctness checking framework, in particular focusing on the collaboration business process models with message exchange or sub-processes [9].

In their previous study [10] authors proposed a method for business process model quality evaluation based on refined Seven Process Modeling Guidelines (7PMG) [11], which resulted in the set of BPMN modeling rules:

- R1: Do not use more than 31 nodes or decompose a model with more than 31 elements if possible (merged G1 and G7);
- R2: Avoid nodes with invalid inputs or outputs (refined G2);
- R3: Avoid usage of multiple start or multiple end events or missing events (G3);
- R4: Avoid gateways mismatch (G4);
- R5: Avoid inclusive (OR) gateways (G5).

As well as the corresponding metrics, thresholds and integrated quality indicators [10]:

- Total number of nodes (TNN);
- Number of invalid elements (NIE);
- Number of start events (NSE);
- Number of end events (NEE);
- Number of mismatched gateways (NMG);
- Total number of gateways (TNG);
- Total number of inclusive (OR) gateways (TNI).

Also, in [12], Beerepoot et al. considered the major BPM problems that should be addressed in the future research. According to [12], authors review nine of the most important research problems in the BPM domain, which require novel solutions and approaches to the analysis, design, and management of business processes with the help of information technologies [12].

In general, considered studies demonstrate the benefits of modeling guidelines and rules in improving the quality of business process models [7], propose methods and frameworks to evaluate BPMN models [8], check their correctness [9], and introduce the importance of quality analysis of created business process models on a continuous basis [10].

2.2. Motivation and Problem Statement

Therefore, the increasing complexity of modern business processes in various organizations requires the development of efficient algorithms and software tools capable of analyzing and correcting error-prone business process models. This will help organizations to achieve better operational performance and avoid errors in their activities caused by incorrect and inefficient business process models of low quality. Especially for critical industries, poor business process models can cause harmful consequences for people, companies, institutions and even nature.

According to Grand View Research [13], the global BPM market size was valued at USD 14.46 billion in 2022 and is expected to grow at a CAGR of 19.9% by 2030 [13], as the international trend towards digitization is increasing day by day.

Moreover, existing studies [2], [3] highlight the challenges of maintaining the correctness of BPMN models, as well as the importance of using design rules to achieve high quality business process models. The continuous quality improvement in business process modeling activities, as well as the emergence of corresponding techniques and tools, is underlined by the evolution of the BPM discipline [4].

Therefore, this paper aims to contribute to BPM practices by addressing the challenges of business process modeling through the algorithmic and software solution for researchers and practitioners to analyze and correct BPMN diagrams.

The structure and functions of the proposed algorithmic and software solution for business process model analysis and correction is demonstrated in Fig. 2.

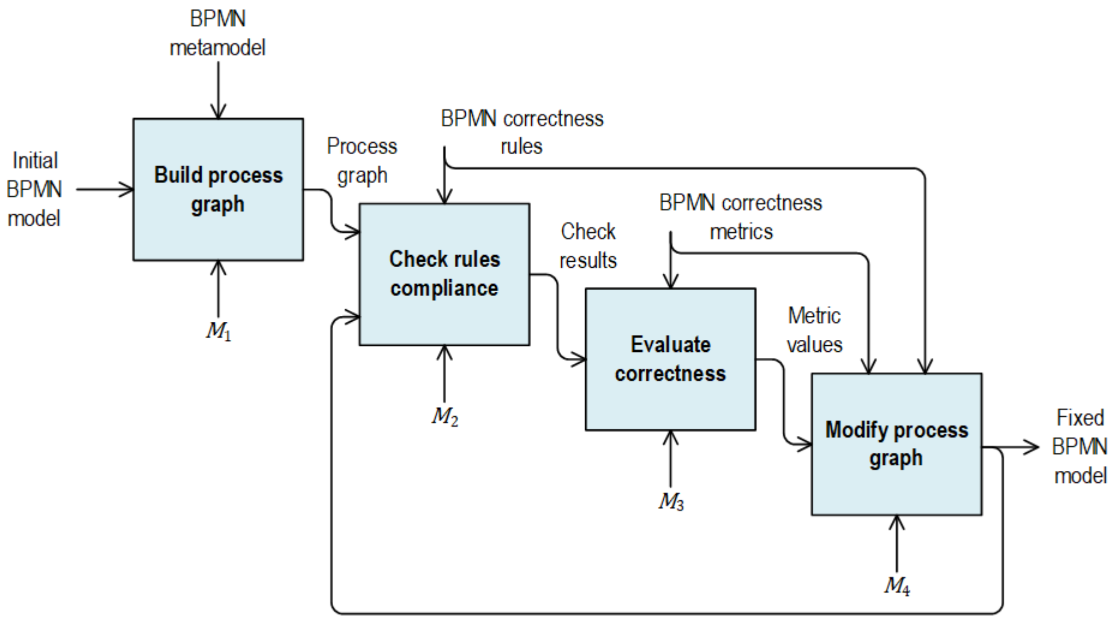

Figure 2: The algorithmic and software solution functions, and structure.

Let us formally describe the proposed algorithmic and software solution (Fig. 2) using the following model [14]:

$$SW = \{M, CF \subseteq M \times M\}, \quad (1)$$

where:

- $M = \{M_1, M_2, M_3, M_4\}$ is the set of interacting software modules;
- CF is the set of control flows between the modules.

The set of interacting software modules represented on the functional design diagram in Fig. 2 should consist of:

- M_1 is a module for creating graph-based structures from the processed BPMN diagrams;
- M_2 is a module for checking the compliance of process structures with modeling rules;
- M_3 is a module for evaluating the correctness of process structures and calculating the related metrics;
- M_4 is a module for modifying process structure elements according to modeling rules and evaluating changes.

The algorithmic foundations of the proposed software modules are discussed in the next Section 3.

3. Materials and Methods

3.1. Formal Business Process Model Representation

Formally the structure of a business process given using the BPMN notation can be described using the directed labeled graph [15]:

$$BP = (N, A, l), \tag{2}$$

where:

- N is the set of nodes that represent various business process elements (i.e. events, activities, gateways, etc.);
- A is the set of arcs that represent business process flows, $A \subseteq N \times N$;
- l is the mapping that assigns textual labels to business process elements and flows:

$$l: N \cup A \to \{\alpha, \alpha \in L\}, \tag{3}$$

where L is the set of textual labels assigned to business process elements and flows; for example, some activity of the order receiving business process could be labeled as "Verify order" to represent the task done by a manager and some flow can be labelled as "Out of stock" to represent one of the scenarios of this business process.

Therefore, let us formally describe a business process model given using the BPMN notation as the following tuple:

$$BPM = \langle m, \Phi, \theta \rangle, \tag{4}$$

where:

- m is the number of business process model elements;
- Φ is the $m \times m$ adjacency matrix [16] of a BPMN business process model, formally represented as the directed labeled graph:

$$\varphi_{ij} = \begin{cases} 1, & (n_i, n_j) \in A, \\ 0, & (n_i, n_j) \notin A, \end{cases} \quad (5)$$

where n_i is the "source" business process element and n_j is the "target" business process element in terms of BPMN flows [17], $i, j = \overline{1, m}$;

- θ is the mapping between business process elements and BPMN element types:

$$\theta: N \to \{SE, EE, BE, IE, TA, GW\}, \quad (6)$$

where:

- SE is a start event;
- EE is an end event;
- BE is a boundary event;
- IE is an intermediate event;
- TA is a task or activity (i.e. a sub-process);
- GW is a gateway.

Using the proposed formal representation of a business process model, it is possible then to validate its elements similarly to graph nodes, evaluate the model's correctness, and provide recommendations for its improvement by searching for corrections in a business process flows matrix Φ.

3.2. Correctness Evaluation of Business Process Models

Most business process modeling software tools follow the BPMN meta-model [17] and prevent connection of elements that should not be connected in a certain way. However, business process model authors still have a freedom to make mistakes when inconsistently modeling events, activities, and gateways making business process models less understandable and, therefore, less efficient.

Therefore, let us formulate the following rules to check whether an element and, hence, a business process model, is correctly modeled or not:

1. If the element is an activity (i.e., a task or a sub-process), it should have one incoming and one outgoing flow.
2. If the element is an intermediate event, it should have one incoming and one outgoing flows.
3. If the element is a gateway, it should have either one incoming and two outgoing flows, or two incoming and one outgoing flow.
4. If the element is a start event, it should not have incoming flows and should have one outgoing flow.
5. If the element is an end event, it should have one incoming flow and should not have outgoing flows.
6. If the element is a boundary event, it should not have incoming flows and should have one outgoing flow.

These rules are based on process modeling "anti-patterns" [18], which include activities that trigger or terminate a workflow instead of start or end events, intermediate events that do the same instead of using the appropriate event types, and gateways that reflect neither splits nor joins.

Hence, let us formally describe rules 1 – 6 given above:

1. Activity rule:

$$r_{TA}(n_i) = \begin{cases} 1, & \delta^+(n_i) = 1 \wedge \delta^-(n_i) = 1, \\ 0, & else. \end{cases} \tag{7}$$

2. Intermediate event rule:

$$r_{IE}(n_i) = \begin{cases} 1, & \delta^+(n_i) = 1 \wedge \delta^-(n_i) = 1, \\ 0, & else. \end{cases} \tag{8}$$

3. Gateway rule:

$$r_{GW}(n_i) = \begin{cases} 1, & (\delta^+(n_i) = 1 \wedge \delta^-(n_i) = 2) \vee (\delta^+(n_i) = 2 \wedge \delta^-(n_i) = 1), \\ 0, & else. \end{cases} \tag{9}$$

4. Start event rule:

$$r_{SE}(n_i) = \begin{cases} 1, & \delta^+(n_i) = 0 \wedge \delta^-(n_i) = 1, \\ 0, & else. \end{cases} \tag{10}$$

5. End event rule:

$$r_{TA}(n_i) = \begin{cases} 1, & \delta^+(n_i) = 1 \wedge \delta^-(n_i) = 0, \\ 0, & else. \end{cases} \tag{11}$$

6. Boundary event rule:

$$r_{TA}(n_i) = \begin{cases} 1, & \delta^+(n_i) = 0 \wedge \delta^-(n_i) = 1, \\ 0, & else. \end{cases} \tag{12}$$

In the formulas above, $\delta^+(n_i)$ and $\delta^-(n_i)$ denote numbers of incoming flows and outgoing flows respectively for the business process element n_i, $i = \overline{1, m}$:

$$\delta^+(n_i) = \sum_{j=1}^{m} \varphi_{ji}, \delta^-(n_i) = \sum_{j=1}^{m} \varphi_{ij}. \tag{13}$$

Therefore, the following formula is used to check if the business process model element is correct or not:

$$r(n_i) = \begin{cases} r_{TA}(n_i), & \theta(n_i) = TA, \\ r_{IE}(n_i), & \theta(n_i) = IE, \\ r_{GW}(n_i), & \theta(n_i) = GW, \\ r_{SE}(n_i), & \theta(n_i) = SE, \\ r_{EE}(n_i), & \theta(n_i) = EE, \\ r_{BE}(n_i), & \theta(n_i) = BE. \end{cases} \quad (14)$$

Finally, we can now introduce general $\rho(N)$ and relative $\sigma(N)$ correctness metrics to estimate the future business process model correction:

$$\rho(N) = \prod_{i=1}^{m} r(n_i)\,, \sigma(N) = \sum_{i=1}^{m} r(n_i). \quad (15)$$

Then, we can use the proposed metrics $\rho(N)$ and $\sigma(N)$ as criteria in the algorithm for business process model correction. This algorithm is expected to provide a new business process flows matrix Φ^* that reveals a new structure of a business process in which all the formulated modeling rules 1 – 6 are followed, such models will be considered as "fixed", i.e. $\rho(N) = 1$. Otherwise, in case the algorithm results in a matrix Φ^*, which allows to satisfy only several modeling rules, such models will be considered as "improved", i.e. $\rho(N) = 0$ but $\sigma(N)$ is increased.

3.3. Business Process Model Correction

Let us propose the algorithm for business process model correction based on the business process flows matrix Φ analysis and correction.

Input: A set of business process elements N, a business process flows matrix Φ.

Output: A modified general correctness metric $\rho^*(N)$, a modified relative correctness metric $\sigma^*(N)$, a modified business process flows matrix Φ^*.

1. Find the initial value of relative correctness metric $\sigma(N)$.
2. Set the iterations counter variable to $\gamma = 0$.
3. Set the maximum number of iterations to the number of invalid elements detected in a BPMN model:

$$\overline{\gamma} = m - \Sigma(N). \quad (16)$$

4. Update the iterations counter variable to $\gamma = \gamma + 1$.
5. For each element φ_{ij}, $i, j = \overline{1, m}$ of the matrix Φ check if $r(n_i) = 0$ or $r(n_j) = 0$, and $i \neq j$ then:

- modify the business process flows matrix element:

$$\varphi_{ij}^* = |1 - \varphi_{ij}|; \quad (17)$$

- re-calculate the value of relative correctness metric $\sigma'(N)$ taking into account the modified business process flows matrix element φ_{ij}^*;

- if $\sigma'(N) \leq \sigma(N)$ return to the previous value of the business process flows matrix element:

$$\varphi_{ij} = \left|1 - \varphi_{ij}^*\right|, \tag{18}$$

otherwise – set $\sigma(N) = \sigma'(N)$.

- If $\gamma = \overline{\gamma}$ stop the algorithm, otherwise – continue to step 4.
- Use the finally obtained modified business process flows matrix Φ^* to find values of modified general correctness metric $\rho^*(N)$ and relative correctness metric $\sigma^*(N)$.

The main idea of the proposed algorithm, is to sequentially verify invalid elements detected in a BPMN model and propose to fix their inefficiencies by adding missing flows between elements. If a certain modification improves the relative correctness metric $\sigma(N)$, i.e., the number of correct nodes is increased to $\sigma'(N)$, the proposed modification φ_{ij}^* is applied to the modified business process flows matrix Φ^*. Otherwise, the modification is cancelled and the next "weak spot" is checked. However, the number of iterations $\overline{\gamma}$ equal to the number of invalid elements in a BPMN model allows to check again "weak spots" for which modifications were cancelled since with the already modified matrix Φ^* such changes can be appropriate and may now improve the correctness of a business process model.

After the matrix Φ^* is produced by the proposed algorithm, it is possible to use it for modifying a BPMN model itself, by adding missing flows between elements. Having the matrix Φ and the modified matrix Φ^*, a flow should be added from the element n_i to the element n_j if $\varphi_{ij}^* > \varphi_{ij}$, $i, j = \overline{1, m}$.

Using the proposed algorithmic foundations, the software tool is designed and developed to perform experiments. The proposed software solution (Fig. 3) is developed using the Python programming language [19] and Python packages to process XML files ("xml" [20]), perform computations ("numpy" [21]), visualize ("matplotlib" [22]), and save research results ("csv" [23]).

4. Results and Discussion

Let us describe the results obtained after applying the developed software tool to experiment with the proposed algorithm for business process model correction. To perform the experiment, we used the public GitHub repository of Camunda [24], which includes 3729 BPMN files:

$$\left\{BPM_k, k = \overline{1, K}\right\}, \tag{19}$$

where k is the index of a business process model from the collection [24] of K BPMN files in total, $K = 3729$.

These models contain business process definitions from different domains: goods dispatch, credit scoring, insurance recourse, and self-service restaurant. There are multiple BPMN models created in these four domains by different people participated in BPMN training sessions [24].

Table 1 below demonstrates performance results of the conducted experiment.

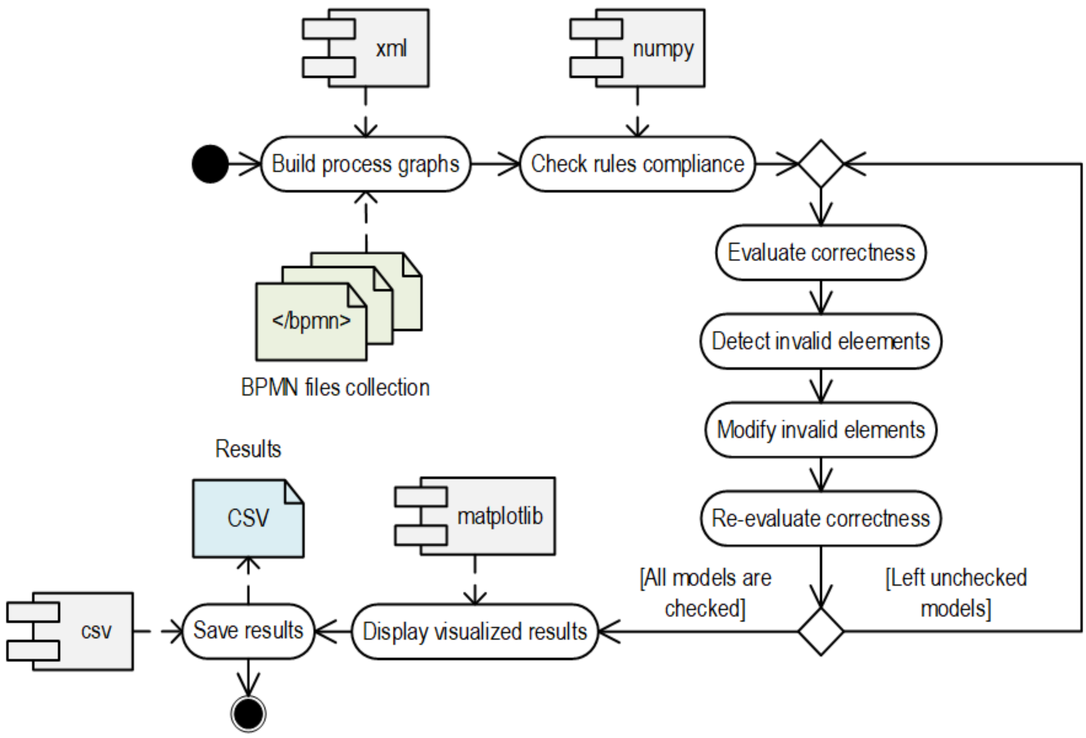

Figure 3: The software solution activity diagram.

Table 1
Performance measures

Measure	Value
Total BPMN models attempted to process	3729
Successfully processed BPMN models	3714 (99.60%)
BPMN models that failed to process	15 (0.40%)
Total time to process BPMN models, sec.	245.74
The average time for processing a BPMN model, sec.	0.15
The standard deviation of the processing time, sec.	0.23

According to Table 1, only a small part of the initial collection of BPMN models [24] was rejected by the software tool as not processable (15 models, 0.40%). This may be related to the presence of syntactic errors in BPMN files. The remaining models (3714 models, 99.60%) were successfully processed by the developed software tool. The total processing time is 245.74 sec. (or only about 4 min.), which is quite a good result for such a large collection of thousands of BPMN models. The average time for processing a BPMN model is 0.15 sec. and the standard deviation of the processing time is 0.23 sec., which means that the proposed algorithm can be used to provide corrections of BPMN models “on the fly”.

Fig. 4 below demonstrates measurements of time required to process each of BPMN models.

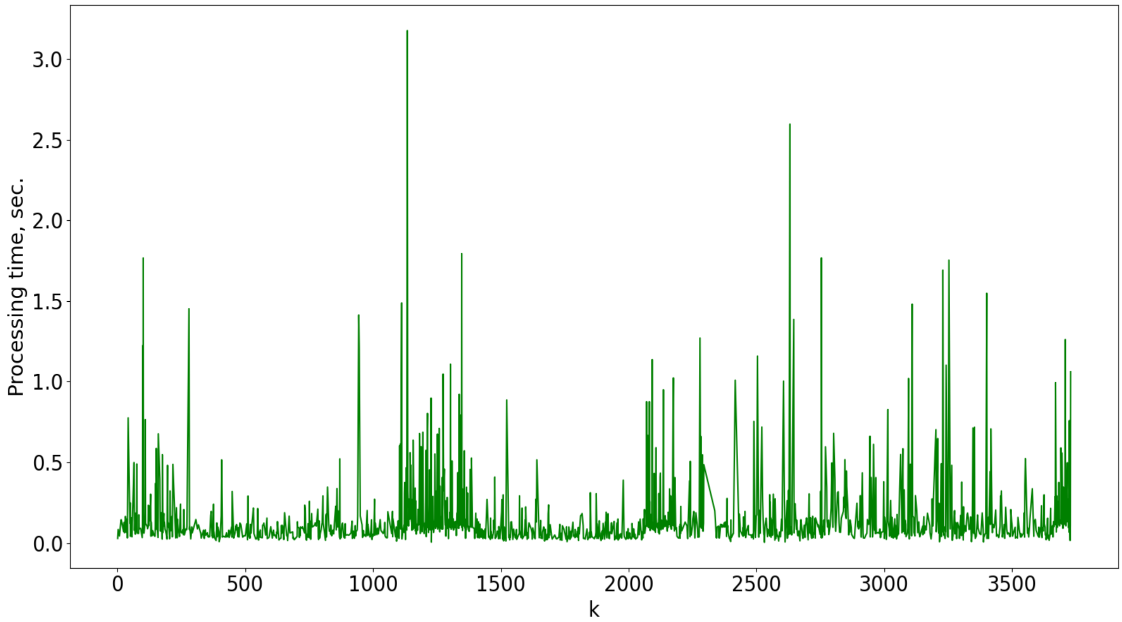

Figure 4: The processing time measurements for analyzed BPMN models.

The exploratory analysis [25] of performance measures results in the following insights:

- the processing time of only 31 BPMN models has exceeded 1 sec. – the processing of 28 models took more than 1 sec., 2 models – more than 2 sec., and 1 model – more than 3 sec.;
- the processing time of 25% of BPMN models is below 0.05 sec., of 50% – below 0.09 sec., and of 75% – below 0.14 sec.;
- the minimum processing time of is 0.004 sec. and the maximum processing time is 3.18 sec.

Table 2 below demonstrates the hardware and software environment used to perform the experiment.

Table 2
Experimental environment

Specification	Property
Processor	AMD A8-7600 Radeon R7, 10 Compute Cores 4C+6G 3.10 GHz
RAM	16.0 GB (15.0 GB usable)
System type	64-bit operating system, x64-based processor
Operating system	Windows 10 Pro 22H2
Runtime environment	Python 3.10 (64-bit)

Table 3 below outlines the results of the business process model evaluation and correction.

Table 3
Evaluation and correction results

Indicator	Value
Invalid models detected, $\sigma(N) < 1$, $\rho(N) = 0$	1641 (44.01%)
Fixed models, $\rho^*(N) > \rho(N)$	1157 (70.63%)
Improved models, $\sigma^*(N) > \sigma(N)$	1293 (78.79%)

The total number of invalid BPMN models is 1641 (44.01%). The number of fixed business process models is 1157 (70.63% of all models detected as invalid), which means that all the violations of rules 1 – 6 were corrected by the proposed algorithm and the new values of general correctness metrics $\rho^*(N)$ for these models were increased to 1, i.e. $\rho^*(N) > \rho(N)$. The number of improved business process models is 1293 (78.79%), which means that some of the violations of rules 1 – 6 were corrected by the proposed algorithm and the new values of relative correctness metrics for these models were increased, i.e. $\sigma^*(N) > \sigma(N)$.

Fig. 5 demonstrates the box plot created using the scaled values of the relative correctness metric obtained before correction $\sigma(N)/m$, after correction $\sigma^*(N)/m$, and the difference between the metric values before and after correction $\sigma^*(N)/m - \sigma(N)/m$.

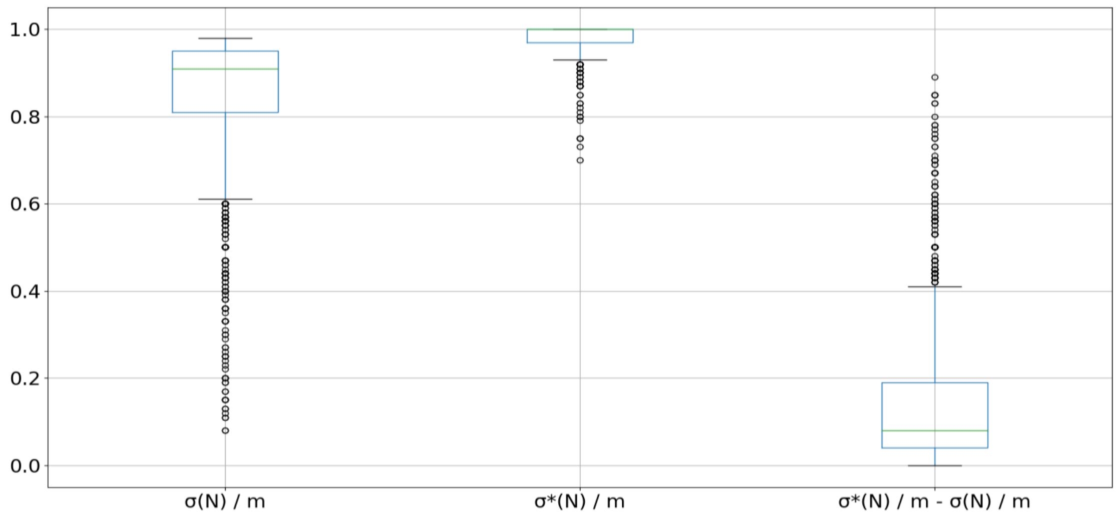

Figure 5: The box blot of relative correctness metric values.

The exploratory analysis [25] of the scaled values of the relative correctness metric results in the following insights demonstrated in Table 4.

Using the results given in Table 4, the following conclusions can be made:

- the mean $\sigma(N)/m$ of analyzed BPMN models is increased by 14% (from 0.85 to 0.99), and the mean of $\sigma^*(N)/m - \sigma(N)/m$ is 0.14 respectively;
- the standard deviation of $\sigma(N)/m$ is decreased by 14% (from 0.16 to 0.02), while the standard deviation of $\sigma^*(N)/m - \sigma(N)/m$ is 0.16 since initially models varied in quality;

- the minimum of $\sigma(N)/m$ is increased by 62% (from 0.08 to 0.70), however, the minimum of $\sigma^*(N)/m - \sigma(N)/m$ is 0.00 since some BPMN models were neither fixed nor improved;
- the lower quartile of $\sigma(N)/m$ is increased by 16% (from 0.81 to 0.97), while the quality of 25% of analyzed BPMN models was improved by 4% (0.04) or less;
- the median of $\sigma(N)/m$ is increased by 9% (from 0.91 to 1.00), while the quality of 50% of analyzed BPMN models was improved by 8% (0.08) or less;
- the upper quartile of $\sigma(N)/m$ is increased by 5% (from 0.95 to 1.00), while the quality of 75% of analyzed BPMN models was improved by 19% (0.19) or less;
- the maximum of $\sigma(N)/m$ is increased by 2% (from 0.98 to 1.00), however, the maximum of $\sigma^*(N)/m - \sigma(N)/m$ is 0.89 for the "worst" analyzed BPMN model improved.

Table 4
Exploratory analysis results of scaled relative correctness metrics

Statistics	$\sigma(N)/m$	$\sigma^*(N)/m$	$\sigma^*(N)/m - \sigma(N)/m$
Mean	0.85	0.99	0.14
Standard deviation	0.16	0.02	0.16
Min	0.08	0.70	0.00
25%	0.81	0.97	0.04
50%	0.91	1.00	0.08
75%	0.95	1.00	0.19
Max	0.98	1.00	0.89

Let us consider the example of a BPMN business process model processing and correction using the proposed algorithmic and software solution. The original BPMN model describes the goods dispatch business process of a small hardware store. It belongs to the Camunda GitHub collection of BPMN files [24].

The graph built using this adjacency matrix Φ and the corresponding BPMN model with the node labels are demonstrated in Fig. 6 below. Red-labelled elements violate rules 1 – 6 and should be fixed.

Fig. 7 demonstrates the corrected business process graph built according to the obtained modified adjacency matrix Φ^* and the corresponding BPMN model with the proposed changes. Green-labelled elements now satisfy business process modeling rule 1 – 6.

As the result, the following BPMN elements were modified to satisfy modeling rules:

- the outgoing flow is added to the start event "Order Received" (SE#01) to connect it with the task "Shipping Clarification" (A#1), $\varphi^*_{1,2} = 1$;
- the outgoing flow is added to the task "Shipping Clarification" (A#1) to connect it with the XOR-gateway (GW#4), $\varphi^*_{2,5} = 1$;
- the outgoing flow is added to the task "Prepare Package to be pick up" (A#12) to connect it with the end event (EE#15), $\varphi^*_{13,16} = 1$.

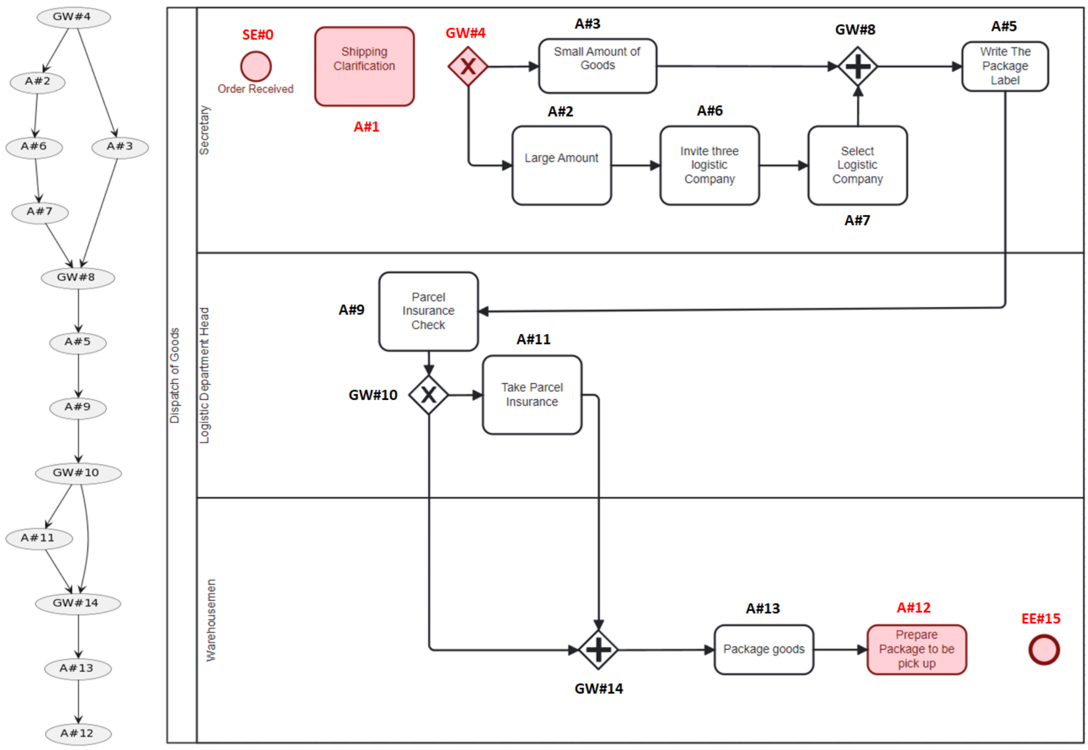

Figure 6: The BPMN model with detected mistakes and its business process graph.

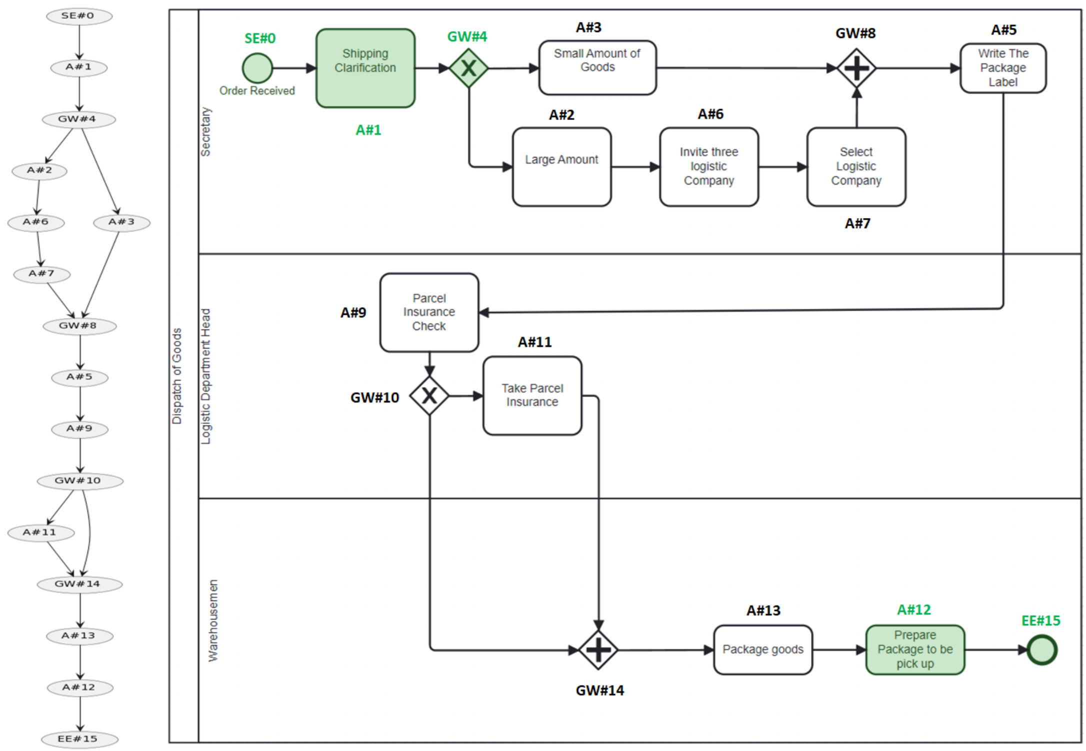

Figure 7: The BPMN model with corrected mistakes and its business process graph.

Let us compare the original (Fig. 6) and modified (Fig. 7) BPMN models using basic process model complexity metrics [5] and introduced correctness metrics (Table 5).

Table 5
Comparison of the original and modified BPMN models

Business process modeling metrics	Original BPMN	Modified BPMN	Change, %
Arcs	14	17	17.65%
Nodes	16	16	0.00%
Coefficient of Network Connectivity (CNC)	0.88	1.06	17.65%
Density	0.06	0.07	17.65%
Interface Complexity (IC)	5184	10000	48.16%
Coupling (CP)	0.06	0.06	0.00%
$\rho(N)$	0	1	100.00%
$\sigma(N)$	11	16	31.25%
$\sigma(N)/m$	0.69	1.00	31.00%

According to Table 5, the formal complexity of the example BPMN model is increased from 17.65% to 48.16% by different complexity metrics [5]. However, this does not mean the understandability of the modified model (Fig. 7) is decreased. In contrast, the missing flows were restored and events, tasks, and a gateway are now properly connected, and demonstrate the consistent workflow. This is confirmed by increased metrics $\rho(N) = 1$, $\sigma(N) = 11$, and $\sigma(N)/m = 1.00$.

5. Conclusions and Future Work

In this study, we have addressed the relevant problem of analyzing and correcting the quality of business process models in order to contribute to the BPM discipline and to face the major challenges in this area, related to the design of understandable and maintainable business process models. In fact, the design of high quality business process models, especially BPMN models as the most widely used workflow diagrams, is one of the success factors for efficient and effective organizational activities. Business process models, which are used to capture, visualize, communicate, and analyze business and information workflows, should be of high quality to avoid errors and mistakes due to their low comprehensibility by both business and information technology professionals in different domains.

Therefore, in this study we analyzed the existing research in the BPM field related to quality analysis of BPMN models and formulated the problem of developing an algorithmic and software solution to ensure continuous quality analysis and correction of the developed business process models. The proposed approach is based on graph-based process description, modeling anti-patterns (i.e. modeling rule violations) detection, and automated correction using the corrections in an adjacency matrix representing the BPMN structure.

The proposed algorithmic foundations were implemented using the Python programming language and additional third-party packages. The software implementation was used to verify the proposed approach. As a test dataset, the collection of 3729 BPMN models was used, in which the maximum processing time of a single BPMN model was 3.18 seconds, indicating the reasonable performance of the proposed solution. In total, the used collection consists of 1641 invalid BPMN models (44.01%), of which 1293 (78.79%) were improved, with 1157 (70.63%) fully corrected to comply with all introduced business process modeling rules 1 - 6.

Future work in this field includes the use of evolutionary optimization algorithms to increase the proportion of fixed business process models and software customization for end users.

References

[1] A. Correia, F. Brito e Abreu, Enhancing the correctness of BPMN models, In: Information Resources Management Association (Ed.), Hershey: IGI Global, 2020, pp. 373–394. doi:110.4018/978-1-5225-9615-8.ch017.

[2] A. Kopp, D. Orlovskyi, Towards Intelligent Technology for Error Detection and Quality Evaluation of Business Process Models, in: IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, Khmelnytskyi, Ukraine, 2023, pp. 1–14. URL: https://ceur-ws.org/Vol-3373/keynote1.pdf.

[3] N. A. Panayiotou, K. E. Stergiou, V. P. Stavrou, Development of a Modeling Architecture Incorporating the Industry 4.0 View for a Company in the Gas Sector, In: Advances in Production Management Systems Towards Smart Production Management Systems: IFIP WG 5.7 International Conference, APMS 2019, Springer International Publishing, 2019, pp. 397–404. doi:10.1007/978-3-030-29996-5_46.

[4] H. A. Reijers, Business Process Management: The evolution of a discipline, Computers in Industry 126 (2021) 103404. doi:10.1016/j.compind.2021.103404.

[5] B. Haj Ayech, S. A. Ghannouchi, E. A. El Hadj Amor, Extension of the BPM lifecycle to promote the maintainability of BPMN models, Procedia Computer Science 181 (2021) 852–860. doi:10.1016/j.procs.2021.01.239.

[6] G. Rivera Lazo, R. Ñanculef, Multi-attribute Transformers for Sequence Prediction in Business Process Management, In: Discovery Science: 25th International Conference, DS 2022, Cham, Springer Nature Switzerland, 2022, pp. 184–194. doi:10.1007/978-3-031-18840-4_14.

[7] C. Fotoglou et al., Complexity clustering of BPMN models: initial experiments with the K-means algorithm, In: Decision Support Systems X: Cognitive Decision Support Systems and Technologies: 6th International Conference on Decision Support System Technology, ICDSST 2020, Springer International Publishing, pp. 57–69. doi:10.1007/978-3-030-46224-6_5.

[8] W. Kbaier, S. A. Ghannouchi, Determining the threshold values of quality metrics in BPMN process models using data mining techniques, Procedia Computer Science 164 (2019) 113–119. doi:10.1016/j.procs.2019.12.161.

[9] F. Corradini et al., Correctness checking for BPMN collaborations with sub-processes, Journal of Systems and Software 166 (2020) 110594. doi:10.1016/j.jss.2020.110594.

[10] A. Kopp, D. Orlovskyi, Towards the Method and Information Technology for Evaluation of Business Process Model Quality, in: ICTERI 2020: Information and Communication Technologies in Education, Research, and Industrial Applications, Springer, Cham, 2021, pp. 93–118. doi:10.1007/978-3-030-77592-6_5.
[11] D. T. Avila, R. I. dos Santos, J. Mendling, L. H. Thom, A systematic literature review of process modeling guidelines and their empirical support, Business Process Management Journal 27(1) (2020) 1–23. doi:10.1108/BPMJ-10-2019-0407.
[12] I. Beerepoot et al., The biggest business process management problems to solve before we die, Computers in Industry 146 (2023) 103837. doi:10.1016/j.compind.2022.103837.
[13] Business Process Management Market Size, Share & Trends Analysis Report By Solution (Automation, Process Modelling), By Application, By Deployment, By End-user, By Region, And Segment Forecasts, 2023 - 2030. URL: https://www.grandviewresearch.com/industry-analysis/business-process-management-bpm-market.
[14] V. L. Lysytskyi, Y. Y. Morhun, Development of software for effective enterprise product policy creation, Bulletin of National Technical University KhPI Series System Analysis Control and Information Technologies 21 (2018) 59–64. doi:10.20998/2079-0023.2018.21.11.
[15] F. Corradini et al., BPMN 2.0 OR-Join Semantics: Global and local characterisation, Information Systems 105 (2022) 101934. doi:10.1016/j.is.2021.101934.
[16] S. Qin, Z. Zhao, J. Su, Z. Yang, The Segmentation and Reconstruction Method of Business Process BPMN under Constraint Conditions, In: 2021 International Conference on Service Science (ICSS), Xi'an, China, 2021, pp. 45–50. doi:10.1109/ICSS53362.2021.00015.
[17] Business Process Model and Notation (BPMN). Version 2.0. URL: https://www.omg.org/spec/BPMN/2.0/PDF.
[18] B. D. Lehmann, P. Alexander, H. Lichter, S. Hacks, Towards the identification of process anti-patterns in enterprise architecture models, In: 8th International Workshop on Quantitative Approaches to Software Quality (QuASoQ), 2020, pp. 47-54. URL: https://ceur-ws.org/Vol-2767/06-QuASoQ-2020.pdf.
[19] Python. URL: https://www.python.org/.
[20] XML Processing Modules. URL: https://docs.python.org/3/library/xml.html.
[21] NumPy. URL: https://numpy.org/.
[22] Matplotlib. URL: https://matplotlib.org/.
[23] CSV File Reading and Writing. URL: https://docs.python.org/3/library/csv.html.
[24] BPMN for research. URL: https://github.com/camunda/bpmn-for-research/.
[25] T. H. Sheikh, P. S. Aithal, Visualization and Explorative Data Analysis, International Journal of Enhanced Research in Science, Technology & Engineering 12(3) 11–21. doi:10.2139/ssrn.4400256.