Method to Recovery Temporal Event Diagram Workflow Model in Computer-Aided Design

Nikolay Voit
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID:0000-0002-4363-4420

Semen Bochkov
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID:0000-0003-1089-4119

Abstract— **In order to strengthen the role of the regulatory documents repository in the form of diagram workflow models (DWFM) of specialists' activities which contains the relation of the events order for the corresponding workflows execution, authors have proposed and investigated a new method to recovery a temporal event DWFM (TEWF) according to the original DWFM. The method is supposed to replace manual labor of specialists in the verification of design, technological and operational errors associated with violations of the events execution order the in the DWFM relevant works with an automated one. The results of TEWF obtaining based on the original DWFM showed a 65% complexity reduction of the original one, depending on the words-concepts glossary.**

Keywords— verification, workflow, diagram model, business process, temporal time, specification, graphic languages, semantics, design

I. Introduction

DWFM obtaining based on design and industrial enterprises specialists' activity 'traces' (designers, technologists and operators) has a wide application in practice of hardware and software CAD activity. These models are viewed as descriptions of the activities process in the form of behavioral models for simulation modeling, increasing information about vulnerabilities to eliminate them [1].

Workflow theory development in CAD and CAM, it solves the inverse problem [2] of DWFM automated design [3–6] based on the fixed design procedures, operations in the form of (history) 'traces' of the designer on completed design tasks events e.g., design protocols, journals, registries, CAD logs.

As a rule, DWFMs are based on visual languages such as UML AD, BPMN, eEPC, IDEF, etc. DWFM of the design, technological and operational process is evaluated during design process analysis by specialists including data mining expert, which consists in process characteristics conditions changing in order to achieve the optimal solution. Partial or full design, technological and operational process interpretation is presented with the DWFM graphic languages (notations). Such models are reusable typical structural units as patterns.

It is worth pointing out that there are plenty of application areas of the project management and workflows theory including human-computer interaction e.g., design production preparation in industry. Modern project process management systems, such as SAP, PeopleSoft, Oracle, CRM software, LOODSMAN: PLM apply the project process management in the form of software technologies, modules. Design process interpretation in such systems is far from a trivial task and requires an understanding of the design process description language (for example, BPMN) and a detailed discussion of the problem with the designers involved in the development process.

Workflow in CAD is useful when analyzing a new system being developed in terms of understanding how the end user (consumer) works with it, herewith the project effects from the workflow in project process management are as follows: time and analysis costs reduction, since most information systems have log modules available; objective information about any accounting, since the DWFM journals are an impartial reflection of the work performed; formal verification of the DWFM properties by interpreting the process before it is put into operation; multiple views support of the same design process using model acquisition and transformation algorithms; involuntary exit from DWFM cycles with the usage of a special acyclic workflow, when necessary, e.g., in case of early cycle break during workflow tracing; multiple changes of DWFM characteristics on check points of a running workflow instance.

In current work, the «time» is well-established concept to state when an event is occurred related to others. In Section 2 there are topic-related works review. In Section 3 the method is described, including RVTI-grammar and a path length calculation procedure to the target vertex $Goal_V$. An experiment in Section 4 is based on a software implementation of the method developed.

In this work, to evaluate the DWFM in a graph form, we will calculate its complexity index with the following formula [7, p. 38]:

$$\rho_1(G) = |E(G)| - |V(G)| + \rho_0(G), \tag{1}$$

where $\rho_1(G)$ is cyclomatic graph number, $\rho_0(G)$ is graph components count; it is also known as «The "zero-th" Betti number of a G» [7, p. 33], «If G is not empty, then, according to theorem 1.18, $\rho_0(G) > 0$. If $\rho_0(G) = 1$, then G is coherent graph; else if $\rho_0(G) > 1$, then G is incoherent graph» (7, p. 33). In current work, $\rho_0(G) = 1$, $|E(G)|$ is edges count, $|V(G)|$ is vertices count.

II. Related works

Process Mining, also known as process discovery or workflow analysis [8], is the discipline of discovering processes from event logs, checking that real processes correspond to idealized ones, and, ultimately, finding ways to improve these processes, including data preprocessing. For the first time such analysis was applied to business processes more than 30 years ago.

Nowadays, there are new algorithms for DWFM obtaining [9]. [10, 11] compare the performance of different algorithms. According to [9], the event log is a main data source in process analysis methods, so log data has a direct impact on the process models quality. Consequently, a low-quality event log e.g., having erroneous, duplicate, missing, or noisy data values, results in a complex, unstructured (spaghetti-like) and process model, which is difficult to be interpreted, or inadequate of the actual business process [9]. In the big data era, the process analysis is severely limited by the event data quality and analysis time. Emamjome et al. [12] present a data quality problem solution by removing the social, material and individual factors that lead to the poor event data quality because such data is not filtered by existing data “filtering” methods. Terekhina et al. [13] study simple Petri nets for workflows modeling, in which OR-split and OR-join transitions without imposed conditions work in the same way as AND-split and AND-join ones. The disadvantage is that the absence of restrictions on uniqueness leads to the construction of an incorrect model. Nestrugina et al. [14] present the workflow optimization of e-document management systems, the prospects for using intelligent tools, methods of data mining. Algorithms which are described in [15] look for anomalies in processes that are limited in structure. A directed graph and cases are considered, when constructing a process model, a trace contains a track of only one process and one trace contains tracks of many processes. Event logs-based processes model building complexity estimation are described. Process analysis extends the functionality of machine learning and data mining by representing data, events, and (manually created or automatically detected) process models in the Petri nets form, BPMN notations, and other models [16].

Well-known theory for assessing the quality of processes is Six Sigma, which contains a set of tools, techniques and methods for improving the process quality [17]. Six Sigma, originally developed by Motorola in the early 1980s and later expanded by many other companies, has an original goal to minimize defects. Van der Aalst [18] presents the Lean Plus Six Sigma project to improve the DWFM quality, which consists of five steps: problem definition and goal setting; key performance indicators measurement and data collection; data analysis to investigate and verify cause-and-effect relationships; analysis-based improvement of current processes; process control to minimize deviations from the target. In the field of process analysis, for education purposes, he also presented a training course [20]. Process detection methods were studied by Ribeiro [18]. He proposed an automated workflow detection service as a recommender system. The quality depends on the set of models and logs used as input. Klaas [19] focused on an overview of the ProM plugins. It should be noted that the labor costs of specialists for verifying descriptions of processes, including data preprocessing, are reduced, which is confirmed in [9, 20, 21], where data preprocessing solution reducing DWFM entropy is described, and, based on it, performance increasing is presented.

Thus, in the above review, there are no automated methods for obtaining a temporal event workflow model from the initial DWFM in CAD, which is a scientific and technical task, the solution of which positively affects the time control of the design tasks execution for the hardware and software systems development.

III. RVTI-grammar

Mathematical support in TEWF synthesis is provided with formal temporal automaton RVTI grammar with internal memory of the language $L\,(G)$ which has the following description: $G = \left(V, \Sigma, \tilde{\Sigma}, C, E, R, \tau, r_0\right)$, where $V = \{v_e, e \in [1, L]\}$ is auxiliary alphabet of operations on internal memory in form of stack $STACK$ or elastic tape, $\Sigma = \left\{a_n{}^{[t_l]},\ n\ \in\ [1, T]\right\}$ is a graphic symbols alphabet, $[t_l]$ can be equal to $\emptyset$, $\tilde{\Sigma} = \left\{\widetilde{a_n{}^{[t_l]}},\ n \in [1, \tilde{T}]\ \right\}$ is a quasi-terms alphabet extended with the graphic objects quasi-terms, graphic objects that are (not) sentential successors, quasi-terms of links-labels with determined semantic differences both for checking graphic objects-continuers and completing verification.

A sentential successor is an element of the quasi-terminal alphabet to neutralize an error in the diagrammatic specification of workflows, which helps to ensure the integrity of the diagrammatic specification of workflows by restoring lost structural links.

$C = \{c, c = c + t_l | \exists t_0 = 0 \rightarrow c = 0\}$ is a counter, $\tau = \{t_l \in [0; +\infty], l \in\ [1, K]\}$ is a time labels set, $c \in [t_l; t_{l+1}]$; $E = \{c{\sim}t_l\}$, where timer c describes conditions of occurrence of the event t_l with relation $\sim\ \in \{=,\ \langle,\ \leq,\ \rangle,\ \geq\}$);

$R = \{r_i, i \in [1, I]\}$ is a grammar scheme, or productions set, where each complex r_i consists of productions subset P_{ij}, $r_i = \left\{P_{ij}, j = \overline{1.J}\right\}$), $r_0 \in R$ is RVTI-grammar axiom, or primary productions complex name, $r_k \in R$ is a finite productions complex. Production $P_{ij} \in r_i$ has the following view:

$$\widetilde{a_l{}^{[t_l]}} \xrightarrow{\Omega_\mu\{W_\gamma(v_1,\ \dots, v_n)/E\}} r_m,$$

where Ω_μ is a memory operation type modifier depending on $\mu\ \in\ \{0,1,2\}$; $W_\gamma(v_1,\ \ \dots,\ \ v_n)$ is n-ary relation determining operation type on internal memory depending on $\gamma = \{1,2,3\}$ (1 is “write”, 2 is “read”, 3 is “compare”), $E = \emptyset$ if c¬~t_l; $(\widetilde{a_l{}^{t_l}})$ are words in form of a couple of quasi-term a_n and temporal label t_l; $r_m \in R$ is succeeding production name.

$P_{ij} \in r_i$ has the following rules: 1. Error – $\nexists P_{lm}$: $\widetilde{a_l{}^{[t_l]}} \xrightarrow{\Omega_\mu\{W_\gamma(v_1, \dots, v_n)/E\}} r_m$; 2. Locking detection – $\sigma = \left\{\widetilde{a_0{}^0}\right\} \rightarrow \left\{\widetilde{a_1{}^{t_1}}\right\} \rightarrow \left\{\widetilde{a_2{}^{t_2}}\right\} \rightarrow \cdots \rightarrow \left\{\widetilde{a_k{}^{t_k}}\right\}$, where $k = 0$; 3. Loop detection – $\sigma = \left\{\widetilde{a_0{}^0}\right\} \rightarrow \left\{\widetilde{a_0{}^{t_0}}\right\}$; 4. Repetition – $\sigma_1 = \sigma_2 = \left\{\widetilde{a_0{}^0}\right\} \rightarrow \left\{\widetilde{a_1{}^{t_1}}\right\} \rightarrow \left\{\widetilde{a_2{}^{t_2}}\right\} \rightarrow \left\{\widetilde{a_k{}^{t_k}}\right\}$; 5. Uncompleted solution and beginning of the new one – $\nexists S_{end} \wedge P_1{}^{TEMP} \prec P_2{}^{TEMP}$; 6. Solution robustness – $\nexists \lim\limits_{k \rightarrow \infty} \left|\left\{\widetilde{a_k{}^{t_k}}\right\}\right| \rightarrow +\infty$. Violations of these rules are semantic and syntactic errors.

The grammar’s language contains words $\widetilde{a_l{}^{t_l}}$, traces $\sigma = \left\{\widetilde{a_0{}^0}\right\} \rightarrow \left\{\widetilde{a_1{}^{t_1}}\right\} \rightarrow \left\{\widetilde{a_2{}^{t_2}}\right\} \rightarrow \cdots \rightarrow \left\{\widetilde{a_k{}^{t_k}}\right\}$ if $E \neq \emptyset$, $\sigma = \{\widetilde{a_0}\} \rightarrow \{\widetilde{a_1}\} \rightarrow \{\widetilde{a_2}\} \rightarrow \cdots \rightarrow \{\widetilde{a_k}\}$ if $E = \emptyset$.

Internal memory can be organized as a stack or store when verifying graphic objects with more than one output to save label links. Elastic tape can be applied to save the number of returns to a graphic object with more than one input, read data from cells and work as a positive integer, or ℕ-based counter.

It is evident that DWFM is input data for TEWF obtaining. It has the following view $Diagram = (V, E, TV, TE)$, where $V = \{v_i, i \in \mathbb{N}\}$ is vertices set; $E = \{e_j, j \in \mathbb{N}\}$ is edges set, $E \subset (V \times V)$; $TV = \{tv_n, n \in \mathbb{N}\}$ is vertices types set; $TE = \{te_m, m \in \mathbb{N}\}$ is edges types set.

$A(RVTI)$ is a finite $RVTI$-based temporal state machine, which controls TEWF synthesis and determines verification rules: $A(RVTI) = (S, T, S_0, P_{ij}, S_{end}, FA)$, where $S = \{s_i, i \in \mathbb{N}\}$ is states set, S_0 is an initial state; T is temporal input graphic words set (diagram terms); $P_{ij} =$

$\left\{\Omega_{\mu}\{W_{\gamma}(v_1,\dots,v_n)/E\}\right\}$ is RVTI-based productions set; where Ω_{μ} is a modifier operator, changing in a certain way memory operation type, $\mu \in \{0,1,2\}$; $W_{\gamma}(v_1,\dots,v_n)$ is n-ary relation determining operation type on internal memory depending on $\gamma = \{1,2,3\}$ (1 is "write", 2 is "read", 3 is "compare"); $E = \{c{\sim}t_l\}$ is a temporal relation, where timer c describes conditions of occurrence of the event t with relation $\sim\, \in \{=, \langle, \leq, \rangle, \geq\}$), $E = \emptyset$ if $c\neg{\sim}t_l$; $FA = \{F_{trans}: S \times T \times P_{ij} \rightarrow S\}$ is transition functions set, S_{end} is finish states set, $F_{start}: V \rightarrow \{0; 1\}$ is an initial vertex search function.

IV. TEWF OBTAINING METHOD

The method to obtain TEWF based on the DWFM contains design procedures shown as functional schemes on Fig. 1 and 2. It reveals sematic patterns (verbs concepts) of DWFM design and technological procedures, presents them on the scale of event-time precedence in the form of TEWF [22, 23].

The method has 4 design procedures described below.

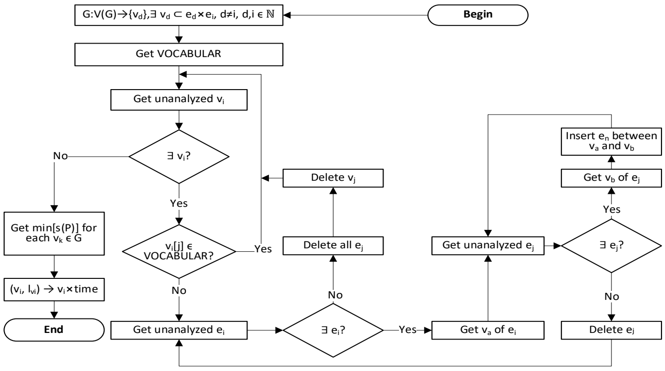
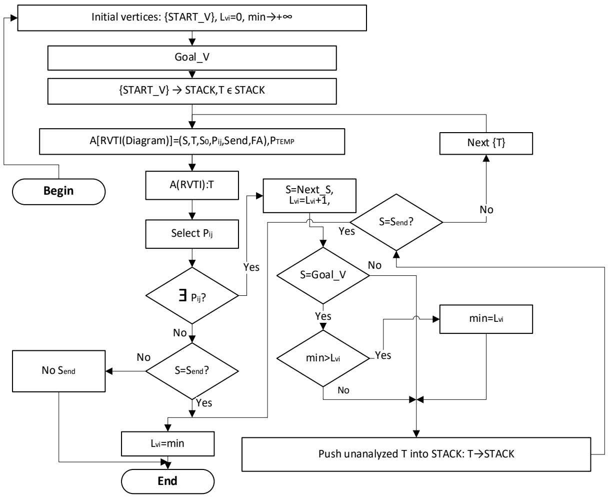


Fig. 1. TEWF obtaining

Procedure 1. Delete vertices v_d without external links e_{di}:

$\nexists v_d \subseteq e_d \times e_i, d \neq i \Rightarrow G{:}\,V(G)\backslash\{v_d\}$, where G is DWFM, $d, i \in \mathbb{N}$

Procedure 2. Hide vertices v_i without verbs-concepts in both project data and title from verbs dictionary $VOCABULAR$ (see Fig. 1).



Initial vertices: {START_V}, L_{vi}=0, min→+∞
Goal_V
{START_V} → STACK,T ∈ STACK
A[RVTI(Diagram)]=(S,T,S_0,P_{ij},Send,FA),P_{TEMP}
Next {T}
Begin
A(RVTI):T
S=Next_S, L_{vi}=L_{vi}+1,
S=S_{end}?
No
Yes
Select P_{ij}
Yes
S=Goal_V
No
∃ P_{ij}?
Yes
min=L_{vi}
No
No S_{end}
No
S=S_{end}?
min>L_{vi}
Yes
No
Yes
L_{vi}=min
Push unanalyzed T into STACK: T→STACK
End

Fig. 2. Counting length of the path to $Goal\_V$

a. Get unanalyzed vertex v_i. If $\nexists v_i$, call **Procedure** 3, else go to b;

b. If v_i contains the verb concept (action) in the design characteristics, go to the next vertex, else go to c;

c. Get unanalyzed incoming directed edge e_i;

d. If $\nexists e_i$, go to h;

e. Get initial vertex v_a of e_i;

f. Get unanalyzed outcoming directed edge e_j;

g. If $\exists e_j$, add directed edge e_n between initial vertex v_a of e_i and ending vertex v_b of e_j, remove e_j else go to i;

h. Go to f;

i. Delete all e_j;

j. Delete v_i, go to a.

Procedure 3. Count minimum length of the path $s(P)$ to each DWFM vertex (see Fig. 2).

The path is $P = (e_1, e_2, \dots, e_n)$, where $n \geq 1$, e_j are DWFM edges. According to [7, p. 171], «n is the length of the path P denoted by $s(P)$», so $n = s(P)$.

Assume $e_1 := e_2 := \dots := e_n := 1$, then path counter is increased by 1 and the procedure of counting length of the path to $Goal\_V$ vertex has the following view.

a. Create initial terms set $START\_V$.

b. Assume path length $l \coloneqq 0$, $l_{min} := +\infty$.

c. Create goal term $Goal\_V$.

d. Push $START\_V$ to $STACK$.

e. Pop an element from $STACK$ and feed it to A.

f. Select a rule in the A matching the current state, the input term, and the transition condition.

   (1) If no rule, go to g, else go to (2).
   (2) $S := Next\_S$.
   (3) Increment l by 1.
   (4) If $S = Goal\_V \land l_{min} > l$, then $l_{min} := l$.
   (5) Push unanalyzed adjacent terms to $STACK$.
   (6) If $Next\_S \in S_{end}$, go to g.
   (7) For the current term, create list of the following terms, feed it to the A and go to (1).

g. Finish. If $S \in S_{end}$ and all terms have been analyzed, output l_{min}, else output a message that the vertex has been failed to achieve probable due to error in DWFM.

This procedure returns a couple of values, namely, vertex and path length (v_i, l_{vi}). It is called as many times as the number of goal vertices, as a rule, for all remaining vertices after the removal.

Procedure 4. Map vertices on the temporal scale in accordance with the counted path lengths: $(v_i, l_{vi}) \rightarrow v_i \times time$, where v_i is vertex matching temporal graphical word of DWFM, l_{vi} is path length from initial vertex to v_i, $time$ is temporal scale in which path lengths are event parameters: $time{:}\, l_{vi} \rightarrow value$, where $value$ is an event index.

V. Experiment

Example of DWFM of design and technological processes and its TEWF are given on Fig. 3 and 4 respectively. Due to pages count limitations for the paper this is the only model presented in details. Table 1 contains the calculated values of the ρ_1 to the original DWFM to the TEWF. Based on the results of 7 DWFMs analysis complexity metrics are calculated on the formula (1). DWFM complexity value is 65% ($\frac{\rho_1(DWFM)-\rho_1(TEWF)}{\rho_1(DWFM)} \cdot 100\%$) depending on dictionary.

TABLE I. Complexity metrics (ρ_1) of DWFM and TEWF

№	**DWFM name**	$\rho_1(DWFM)$	$\rho_1(TEWF)$
1	Initial stage of the product design	9	1
2	Labor intensity	2	0
3	Technological processes of the large industrial enterprise	6	2
4	Design specification formation	8	0
5	Design specification formation (in details)	8	0
6	Technical task creation	6	1
7	CAD/CAM/CAPP	4	11
Average		$\mathbf{6\frac{1}{7}}$	$\mathbf{2\frac{1}{7}}$

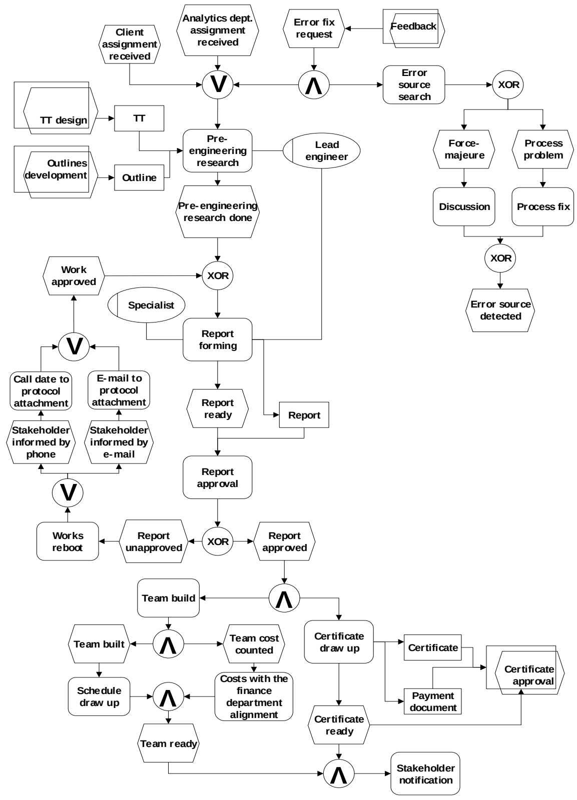


Fig. 4. DWFM “Initial stage of the product design” ($|E(G)| = 62, |V(G)| = 54$). TT is for *Technical Task*

Conclusion

In the paper, authors have presented a new method for strengthening the role of the normative documents base in the form of a DWFM, containing the events execution order relation in the relevant work and preliminary processing workflow data. It provides obtaining a TEWF from the original DWFM which event time-ordered temporal graphic words-concepts is, extends the Workflow Theory [24], reduces the DWFM complexity to 65% of the original one according to experiments, depending on the dictionary of concepts. The method is distinguished by the ‘time’ concept usage that identifies errors associated with the synchronization and parallelism of temporal graphic words in the computer-aided design of radio electronics hardware and software. The method contributes to the automation of the specialists’ labor to detect errors which violate requirements for the events order of relevant works. DWFM complexity reduction refers to data preprocessing and, according to [9, 20, 21], helps to reduce the labor intensity of their verification by specialists by reducing their entropy.

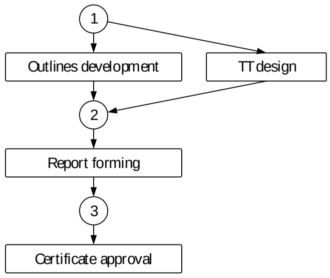


Fig. 3. TEWF “Initial stage of the product design” ($|E(G)| = 7, |V(G)| = 7$)

The proposed method is used at such a large radio-electronic design and industrial enterprise as JSC ‘Ulyanovsk Mechanical Plant’ (Ulyanovsk, Russian Federation), particularly, in the hardware and software development for the products of the enterprise.

References

[1] Special Report, https://smotrim.ru/brand/5410, last accessed 2023/04/03.

[2] Tarantola, A: Inverse problem theory and methods for model parameter estimation. Society for industrial and applied mathematics, (2005).

[3] Singh, G. N., Sandeep A.: A process model for workflow mining. International Journal of Information Technology and Knowledge Management 4(2), 719–722 (2011).

[4] Van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Which processes can be rediscovered? Beta publicatie: working paper series 74, 1–25 (2002).

[5] Van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering 16(9), 1128–1142 (2004).

[6] Hammori, M., Herbst, J., Kleiner, N.: Interactive workflow mining requirements, concepts and implementation. Data & Knowledge Engineering 56(1), 41–63 (2006).

[7] Tutte, W.: Graph Theory. Addison-Wesley, CA, USA (1984).

[8] Williams, R., Rojas, E., Peek, N., Johnson, O.: Process mining in primary care: a literature review. Studies in health technology and informatics 247, 376–380 (2018).

[9] Marin-Castro, H. M., Tello-Leal, E.: Event log preprocessing for process mining: a review. Applied Sciences 11(22) (2021).

[10] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi F.M., Marrella, A., Mecella, M., Soo, A.: Automated Discovery of Process Models from Event Logs: Review and Benchmark. IEEE Transactions on Knowledge and Data Engineering 31(4), 686–705 (2017).

[11] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A Multi-Dimensional Quality Assessment of State-of-the-Art Process Discovery Algorithms Using Real-Life Event Logs. Information Systems 37(7), 654–676 (2012).

[12] Emamjome, F., Andrews, R., ter Hofstede, A., Reijers, H.: Alohomora: Unlocking data quality causes through event log context. In: Proceedings of the 28th European Conference on Information Systems (ECIS2020), Marrakech, Morocco, 15–17 June 2020, pp. 1–16. Association for Information Systems, Atlanta, GA, USA (2020).

[13] Terekhina, E., Grusho, A., Timonina, E., Shorgin, S.: Constructing Process Models Represented by Simple Petri Nets. Systems and Means of Informatics 30(4), 61–75 (2020).

[14] Nestrugina, E., Zaitseva, A.: Intelligent data analysis in electronic electronic document management. In: Pavlysh, V.N. (ed.) Proceedings of 7th International Conference on Modern Information Technologies in Education and Scientific Researches (SITONI-2021), pp. 473–477. Donetsk: Donetsk National Technical University (2021).

[15] Teryokhina, I.: Anomaly detection in several running processes. International Journal of Open Information Technologies 10(1), 21–27 (2022).

[16] Van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer-Verlag, Berlin, 2016.

[17] Pyzdek, T.: The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts, and Managers at All Levels. McGraw Hill, New York, USA (2003).

[18] Ribeiro, J., Carmona, J., Mısır, M., Sebag, M.: A Recommender System for Process Discovery. In: Sadiq, S., Soffer, P., Voelzer, H. (eds.) International Conference on Business Process Management (BPM 2014), LNCS, vol. 8659, pp. 67–83. Springer-Verlag, Berlin (2014).

[19] Claes, J., Poels, G.: Process Mining and the ProM Framework: An Exploratory Survey. In: La Rosa, M., Soffer, P. (eds) Business Process Management Workshops. BPM 2012. Lecture Notes in Business Information Processing, vol. 132, pp. 187–198. Springer, Berlin, Heidelberg (2013).

[20] Van der Aalst, W.: Process Mining: Data science in Action, https://www.coursera.org/learn/process-mining/supplement/56T4T/quick-note-regarding-quizzes-in-this-course, last accessed 2023/04/03.

[21] Bose, R.P.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna improve process mining results? In: Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 127–134. Singapore (2013).

[22] N. Voit and S. Bochkov: Verification Method for Diagrammatic Workflow Specifications in Computer-Aided Design. In: 2022 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, pp. 188-193, doi: 10.1109/RusAutoCon54946.2022.9896254. (2022).

[23] Voit, N., & Bochkov, S.: Denotative and significative semantics analysis and control methods in diagrams. CEUR Workshop Proceedings, 2790, 19-32 (2020).

[24] Demmenie, G.P.: Workflow Mining: A stepwise approach for extracting Event logs from Corporate Network Data. MSc Thesis. Delft University of Technology, Delft, Netherlands (2011).