https://doi.org/10.31891/csit-2023-3-5
UDC 004.94

Oleksandr RUDSKYI, Andrii KOPP
National Technical University «Kharkiv Polytechnic Institute»

ALGORITHM AND SOFTWARE TO ASSESSS THE COMPLIANCE OF BUSINESS PROCESS MODELS WITH THEIR TEXTUAL DESCRIPTIONS

This paper is devoted to solve the problem of analyzing the compliance of business process models with their textual descriptions. The problem being business process models describe re-designed or completely new organizational activities, but "wrong" models that do not reflect correctly business process requirements may mislead involved business process participants and other stakeholders, and cause workflow errors followed by extra costs. Therefore, the research goal is to ensure the correctness of business process models by analyzing their compliance with textual descriptions formulated by business process owners or business analysts. In the work, a review of existing tools for modeling and analysis of business processes is outlined, as well as the main technologies of natural language processing are considered, including tokenization, search for stop words, and stemming. These technologies are proposed to be used to analyze the compliance of business process models with their textual descriptions. An approach to solving the problem of analyzing the compliance of business process models with their textual descriptions, using the selected natural language processing tools is proposed and the respective algorithm is developed. The process of analyzing the compliance of business process models with their textual descriptions is formalized using data flow modeling. The corresponding software that implements this process is developed. Sample calculations are demonstrated that confirm the performance of the proposed approach by analyzing the model of the goods dispatch business process and the corresponding textual description of this business process. Finally, conclusions are given and the directions for further work are determined.

In the future, it is necessary to elaborate the software that will help business users analyze the compliance of BPMN models with textual descriptions of depicted workflows, as well as to elaborate the developed algorithm by using advanced artificial intelligence methods, e.g. neural networks, trained on the collection of real business process models.

Keywords: business process model analysis, business process model compliance with textual description, natural language processing, analysis of activity text labels.

Олександр РУДСЬКИЙ, Андрій КОПП
Національний технічний університет «Харківський політехнічний інститут»

АЛГОРИТМ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ВІДПОВІДНОСТІ МОДЕЛЕЙ БІЗНЕС-ПРОЦЕСІВ ЇХ ТЕКСТОВИМ ОПИСАМ

Дана робота присвячена вирішенню проблеми аналізу відповідності моделей бізнес-процесів їх текстовим описам. Проблема полягає в тому, що моделі бізнес-процесів описують перепроектовані або абсолютно нові процеси діяльності організації, але "неправильні" моделі, які некоректно відображають вимоги до процесів, можуть вводити в оману учасників бізнес-процесів та інших зацікавлених сторін, а також спричиняти помилки в ході виконання бізнес-процесів, що призводять до додаткових витрат. Тому метою дослідження є забезпечення коректності моделей бізнес-процесів шляхом аналізу їх відповідності текстовим описам, сформульованим власниками бізнес-процесів або бізнес-аналітиками. У роботі здійснено огляд існуючих інструментів моделювання та аналізу бізнес-процесів, а також розглянуто основні технології обробки природної мови, серед яких токенізація, пошук стоп-слів та стеммінг. Ці технології пропонується використовувати для аналізу відповідності моделей бізнес-процесів їх текстовим описам. Запропоновано підхід до розв'язання задачі аналізу відповідності моделей бізнес-процесів їх текстовим описам з використанням обраних засобів обробки природної мови та розроблено відповідний алгоритм. Формалізовано процес аналізу відповідності моделей бізнес-процесів їх текстовим описам за допомогою моделювання потоків даних та розроблено відповідне програмне забезпечення, яке реалізує цей процес. Продемонстровано приклади розрахунків, які підтверджують працездатність запропонованого підходу на прикладі аналізу моделі бізнес-процесу доставки продукції та відповідного текстового опису даного бізнес-процесу. На завершення зроблено висновки та визначено напрямки подальших досліджень.

У майбутньому необхідно розробити програмне забезпечення, яке допоможе бізнес-користувачам аналізувати відповідність BPMN-моделей текстовим описам зображених робочих процесів, а також доопрацювати розроблений алгоритм за допомогою сучасних методів штучного інтелекту, наприклад, нейронних мереж, навчених на колекції реальних моделей бізнес-процесів.

Ключові слова: аналіз моделей бізнес-процесів, відповідність моделі бізнес-процесів текстовому опису, обробка природної мови, аналіз текстових міток дій.

Introduction

Business process models have proven themselves to be effective tools for the visualization and improvement of complex organizational operations. Business process models are used to find inefficient places in the described business processes and to eliminate the identified shortcomings by automation with the help of customizable software solutions or unified software environments for the execution of business processes [1].

BPM (Business Process Management) is the concept of managing an organization at the level of processes, which are considered as a business resource that is constantly changing and adapting to changes within and in the environment. The main principles of this concept are transparency and comprehensibility of business processes. To achieve this goal, process modeling is resorted to using notations of a certain stable standard. The most common standard of such notation is BPMN 2.0 (Business Process Model and Notation) [2]. The notation defines a constant list of elements that are used to build diagrams – models of business processes. Such models, as a rule, include a set of actions and events.

However, the creation of business process models is a time-consuming task that requires significant human resources, so there may be situations in which the business process model does not correspond to the textual description of the business process it is supposed to represent. This can lead to errors in the execution of the business process, loss of time, and, accordingly, unforeseen monetary costs.

Thus, the task of analyzing the compliance of business process models with their textual descriptions is relevant [1] and requires the usage of computational intelligence methods and techniques to estimate the correspondence of BPMN models to workflow descriptions, and assume business process models' correctness and adequacy.

Review of existing business process modeling and analysis solutions

The first among the considered tools is Bonita BPM. Bonita BPM is a business process management software. This software that should be installed on the computer of an analyst or developer. With the help of this tool, users can edit BPMN diagrams, create data models, download user manuals, and design forms [3].

Next among the analogs is another application – Signavio. Signavio Process Manager is a web-based solution for design, analysis, and documenting (modeling) of business processes. This solution allows to create process models as the flowcharts directly in the browser, link any document with a business processes (work procedures, regulations, instructions for the provision of services, etc.), document decisions within processes in a graphical form, export processes in various formats ("png", "svg", "pdf", "xml" for BPMN 2.0) [4].

Another alternative is ProcessMaker, which is the software for business process management. It allows users to effectively model their business processes. The software is fully accessible over the Internet and accessible through any web browser, simplifying the management and coordination of business processes throughout the organization [5].

Bizagi is the software for building business process maps and models in BPMN notation. It allows users to create, interpret, and optimize workflow diagrams using BPMN notation, and publish business process documentation in Word, PDF, Excel, and Wiki formats [6].

As for the brief conclusion on the considered business process modeling and analysis tools, we can formulate the following:

1) all of the considered software tools are relatively easy to use and require only the knowledge of BPMN notation without special training in information technologies;

2) Signavio Process Manager and ProcessMaker are web-based tools, so users do not need to download and install the application;

3) Bonita BPM and Bizagi are less convenient since users have to install the software on their workstations;

4) none of these most well-known and widely-used software tools for business process modeling and analysis allow checking the compliance of BPMN models with the initial descriptions of depicted business processes.

After the analysis of the existing software tools, it can be concluded that all applications greatly simplify the process of building business process models, but these applications do not provide an opportunity to analyze the models for errors. First of all, they do not have the opportunity to compare business process models with their textual descriptions, which can lead to errors in the execution of the business process, loss of time, and significant financial costs when implementing models that are inadequate to the subject area.

Suggested natural language processing techniques

In order to analyze the compliance of business process models with their textual descriptions, it is suggested to apply the following NLP (Natural Language Processing) technologies, briefly described in Fig. 1:

1) tokenization;

2) search for stop words;

3) stemming.

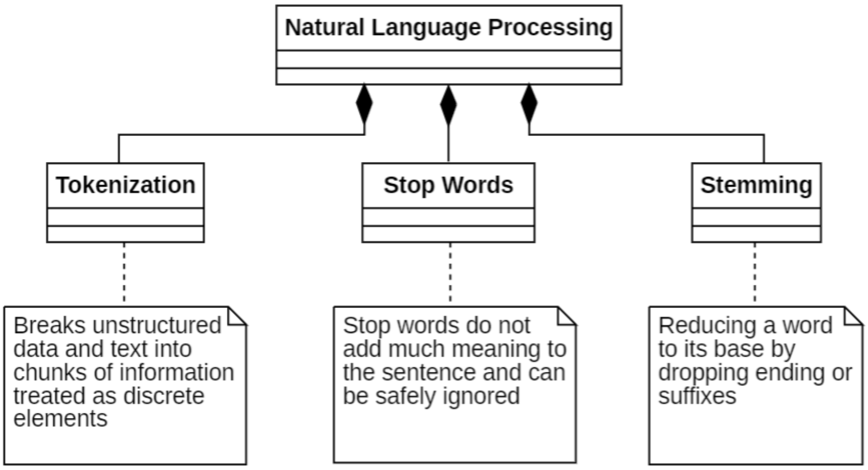

Fig. 1. Considered NLP techniques

Tokenization is the first step in any NLP process. The tokenizer breaks unstructured data and text written in natural language into blocks of information that can be treated as discrete elements. It allows to convert an unstructured string (a text document) into a numerical data structure suitable for machine learning. Tokenization can be used directly as a separate operation or in the process of machine learning as a step followed by more complex actions [7].

Tokenization can be performed for a word, a symbol, or a part of a word [8]:

1) word tokenization is the most commonly used tokenization algorithm, which breaks a piece of text into separate words, taking into account a certain separator; depending on the separators, a variety of word-level markers are formed [8];

2) character tokenization breaks a part of the text into a set of characters [8];

3) tokenization of subwords breaks a text fragment into subwords (or n-gram symbols); for example, such words as “lower” can be segmented as “low-er”, “smartest” as “smart-est” and so on [8].

Once the text is tokenized, it is often clear that not all words carry the same amount of information, if any. Common words that carry little meaningful information are called stop words. Stop words are words in any language that do not add significant meaning to a sentence. We should ignore them without sacrificing the meaning of the sentence. These are some of the most common, short function words such as “the”, “is”, “at”, “which” and “on”. In this case, stop words can cause problems when searching for phrases that include them [9].

If there is a task of text classification or tonality analysis, stop words should be removed because they do not provide any information to the model. That is, to exclude unwanted words from the corpus. But if there is language translation, then stop words should be left, since they should be translated together with other words [9].

Stemming is one of the most common data preprocessing operations performed in almost all NLP projects. Stemming is the process of reducing a word to its base by dropping auxiliary parts such as endings or suffixes. The results of stemming are sometimes very similar to determining the root of a word, but its algorithms are based on different principles. Therefore, the word after processing by the stemming algorithm may differ from the morphological root of the word [10].

There are several variants of stemming algorithms, which differ in their accuracy and performance:

1) search by a table – this algorithm uses a principle of searching by a table in which all possible variants of words and their forms after stemming are collected [10];

2) cutting off endings and suffixes – these algorithms are based on the rules according to which a word can be shortened [10];

3) lemmatization is a more complex approach based on determining the base of the word through lemmatization; the first step of this algorithm is the determination of parts of speech (POS) in a sentence, i.e. “POS tagging”, in the second step, stemming rules are applied to the word according to the part of speech [10];

4) stochastic algorithms – these algorithms are based on the probability of determining the basis of a word [10];

5) hybrid approach – when building a hybrid stemming algorithm, a combination of the above algorithms can be used; for example, the algorithm can use the method of cutting off endings and suffixes, but at the first stage perform a table search [10];

6) matching search – these algorithms use a knowledge base that contains only the bases of words, that is, this knowledge base consists of those words into which ordinary words are transformed after stemming [10].

Object, subject, and methods of research

Existing software tools provide for the construction of business process models but do not provide an opportunity to analyze these models from the point of view of their adequacy to real business processes, namely, to compare business process models with their textual description. Therefore, the development of an approach to the analysis of the compliance of business process models with their textual descriptions is relevant.

The research object is the process of analyzing the compliance of business process models with their textual descriptions. The research subject is an algorithm for analyzing the compliance of business process models with their textual descriptions. The research purpose is to ensure the adequacy of business process models by analyzing their compliance with textual descriptions.

Thus, it is necessary to form the text T_1 from all names of elements of type "task" and related elements of actions. The following algorithm that can be used for this task is represented by the UML activity diagram in Fig. 2.

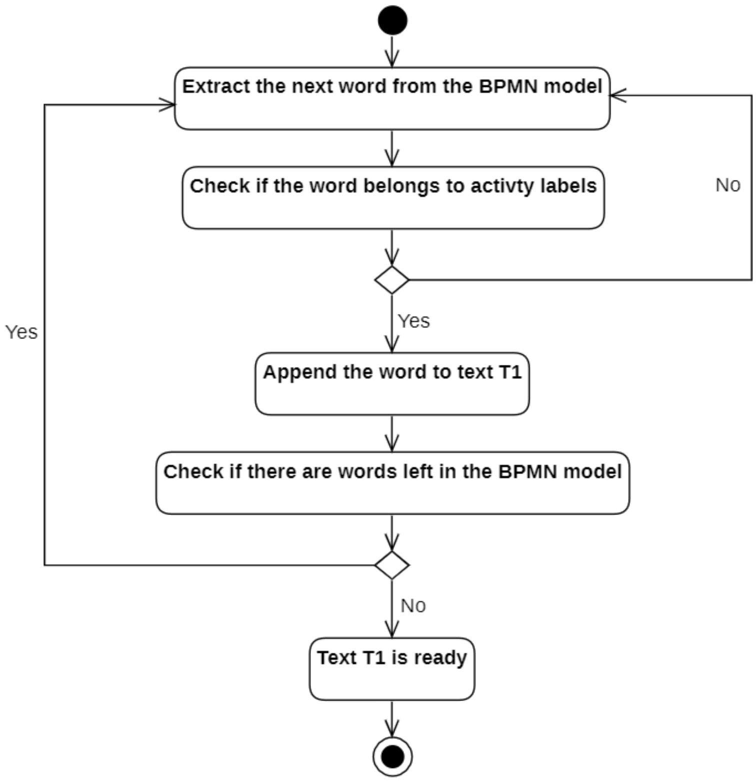

Fig. 2. Algorithm for generating text from task names of the business process model

The algorithm and respective software should extract from the BPMN file all the names of elements of type "task" and related elements of actions:

1) "Service Task" is a task that uses some service, which can be a web service or an automated application [11];

2) "Send Task" is a simple task designed to send a message to an external participant. As soon as the message is sent, the task is completed [11];

3) "Receive Task" is a simple task designed to wait for receiving a message from an external user [11];

4) "User Task" is a typical task of a business process in which a human executor performs a task with the help of a software application and is scheduled through some task list manager [11];

5) "Manual Task" is a task that is supposed to be performed without the help of any business process execution mechanism or any program [11];

6) "Business Rule Task" is a task that provides a mechanical process to provide input data for the business rule mechanism and obtain the output data of calculations that the business rule mechanism can provide [11];

7) "Script Task" is a task that is executed by the business process engine. When the task is ready to run, the engine will execute the script. After completing the script, the task will also be executed [11].

Along with the BPMN file of the business process model, the input is also a textual description T_2 of the business process, which this model should represent.

Thus, to solve the problem of analyzing the compliance of business process models with their text descriptions using the received texts T_1 and T_2, the following actions must be performed:

1) split the input texts T_1 and T_2 into separate words (tokenize) and obtain the corresponding multisets of words:

$$(W_1, m_1) = \left\{\left(t_i^1, m_1(t_i^1)\right), t_i^1 \in W_1 \wedge i = \overline{1,n}\right\},$$
$$(W_2, m_2) = \left\{\left(t_j^2, m_2(t_j^2)\right), t_j^2 \in W_2 \wedge j = \overline{1,q}\right\}, \quad (1)$$

where:

– W_1 is the set of words obtained as a result of the tokenization of text T_1;

– W_2 is a set of words obtained as a result of the tokenization of text T_2;

– $t_i^1 \in W_1,\ i = \overline{1,n}$ – word obtained as a result of tokenization of text T_1;

– $t_j^2 \in W_2,\ j = \overline{1,q}$ – word obtained as a result of tokenization of text T_2;

– $m_1(t_i^1)$ – mapping $m_1: W_1 \to \mathbb{Z}^+$, which for each word $t_i^1 \in W_1,\ i = \overline{1,n}$ sets the number of its repetitions in the text T_1;

– $m_2(t_j^2)$ – mapping $m_2: W_2 \to \mathbb{Z}^+$, which for each word $t_j^2 \in W_2,\ j = \overline{1,q}$ sets the number of its repetitions in the text T_2;

– n is the number of words obtained as a result of the tokenization of text T_1;

– q is the number of words obtained as a result of the tokenization of text T_2;

2) remove stop words from sets W_1 and W_2 to obtain sets of only meaningful terms related to the subject area of the business process:

$$stop: \{W_k, k = \overline{1,r}\} \to \{W'_k, k = \overline{1,r}\}, \quad (2)$$

where:

– $W_k,\ k = \overline{1,r}$ is the set of words obtained as a result of tokenization of the source text, which also contains stop words;

– $W'_k,\ k = \overline{1,r}$ – set of words obtained as a result of tokenization of the source text, from which stop words are removed;

– $stop$ is a mapping that, for each set $W_k,\ k = \overline{1,r}$, which contains stop words, matches the set $W'_k,\ k = \overline{1,r}$, which does not contain stop words;

– r is the number of sets of words processed, $r = 2$;

3) perform stemming of words in sets W'_1 and W'_2, remaining after removing stop words:

$$stemm: \{W'_k, k = \overline{1,r}\} \to \{W''_k, k = \overline{1,r}\}, \quad (3)$$

where:

– $W''_k,\ k = \overline{1,r}$ is the set of words obtained as a result of stemming the words remaining after the removal of stop words;

– $stemm$ is a mapping which, for each set $W'_k,\ k = \overline{1,r}$, from which the stop words are removed, matches the set $W''_k,\ k = \overline{1,r}$ in which words remained after the removal of the stop words, are shortened to the base.

Thus, as a result of the previous actions (1) – (3), two sets of words W''_1 and W''_2 will be obtained:

$$W''_1, W''_2 \in \{W''_k, k = \overline{1,r}\}, \quad (4)$$

where:

– W''_1 is the set of words obtained after the processing steps (1) – (3) are completed for the text T_1 built using all names of elements of type “task” and related elements of BPMN business process model actions;

– W''_2 is a set of words obtained after the processing steps (1) – (3) are completed for the textual description T_2 of the business process, which the BPMN model should represent.

The similarity of these two sets of words W''_1 and W''_2 (4) can be calculated using one of the distance metrics [12]. The Jacquard coefficient is suggested for use in the problem-solving algorithm because it gives an accurate estimate in the range between 0 and 1, and at the same time is simple to implement.

Hence, the obtained value of the Jaccard coefficient [12] can be interpreted as the degree of compliance of the business process model with its textual description:

$$K_J = \frac{|W''_1 \cap W''_2|}{|W''_1| + |W''_2| - |W''_1 \cap W''_2|}. \quad (5)$$

Thus, the paper elaborates the following algorithm, earlier proposed by authors in [13], for solving the task of analyzing the compliance of business process models with their textual descriptions, represented by the UML activity diagram in Fig. 3.

Since the Jaccard similarity coefficient [12] produces values in the 0 – 1 range, it is proposed to measure the degree of correspondence of BPMN workflow diagrams to the initial textual descriptions of ongoing or planned business processes. For this purpose, we suggest using the psychophysical Harrington’s scale of quality [14]:

1) 0.00 – 0.20 for the “very bad” compliance;
2) 0.21 – 0.37 for the “bad” compliance;
3) 0.38 – 0.63 for the “satisfactory” compliance;
4) 0.64 – 0.80 for the “good” compliance;
5) 0.81 – 1.00 for the “very good” compliance.

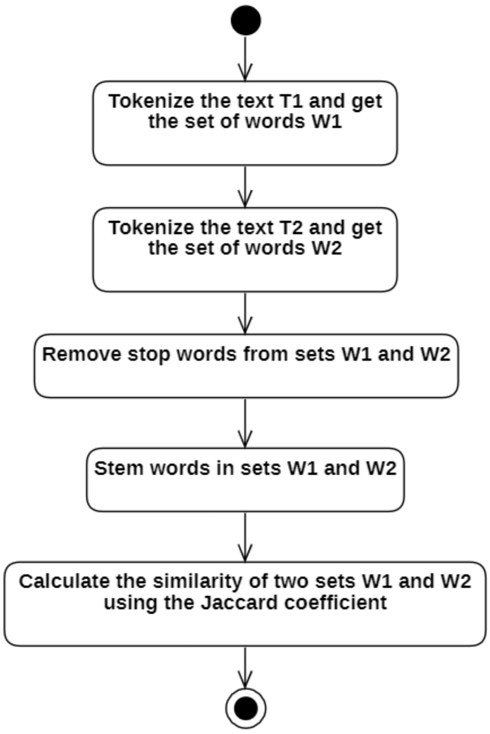

Fig. 3. Algorithm for solving the problem of analyzing the compliance of business process models with their textual descriptions

Results and discussion

In the process of analyzing the compliance of business process models with their text descriptions, the BPMN file of the model and the corresponding text description of the real business process are used. Compliance check is carried out on the basis of the developed algorithm. Future software based on the proposed algorithm should generate a report based on the verification result. It is intended that the software will be used by both registered and non-registered users. The software requirements were earlier formulated by authors in [15].

Using the DFD (Data Flow Diagram) modeling method, a context diagram of the process of business process models compliance analysis with their textual descriptions is designed, which is shown in Fig. 4.

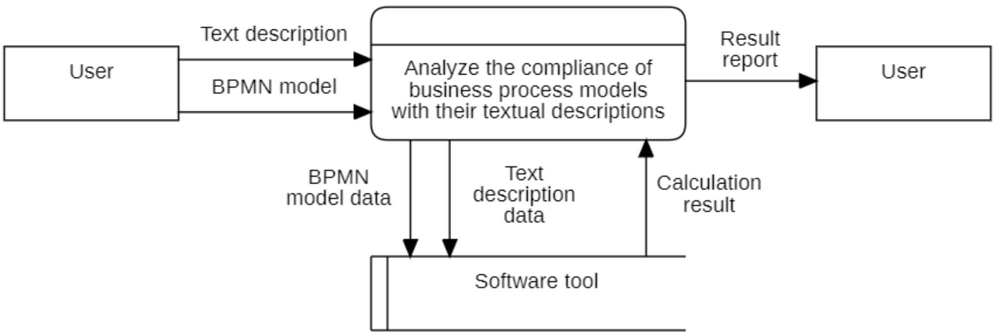

Fig. 4. Process context diagram

This diagram (Fig. 4) shows inputs and an output, the software tool implementing the algorithm given in Fig. 3, and involved participants (users) that analyze the compliance of business process models with their textual descriptions.

A decomposition diagram of the process of analyzing the compliance of business process models with their textual descriptions is designed, which is shown in Fig. 5.

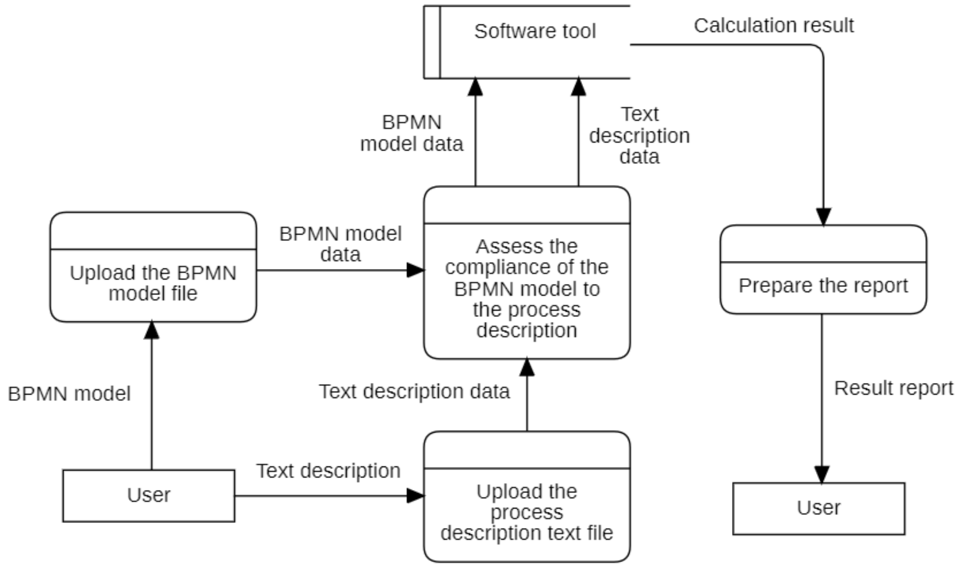

Fig. 5. Process decomposition diagram

This diagram (Fig. 5) shows the activities of BPMN models and text descriptions uploading, analysis of the compliance of business process models with their textual descriptions using the proposed algorithm (Fig. 3), and report generation with the provided evaluation and recommendations.

Let us consider the sample business process model of the goods dispatch process that happens in a small hardware store (Fig. 6) [16].

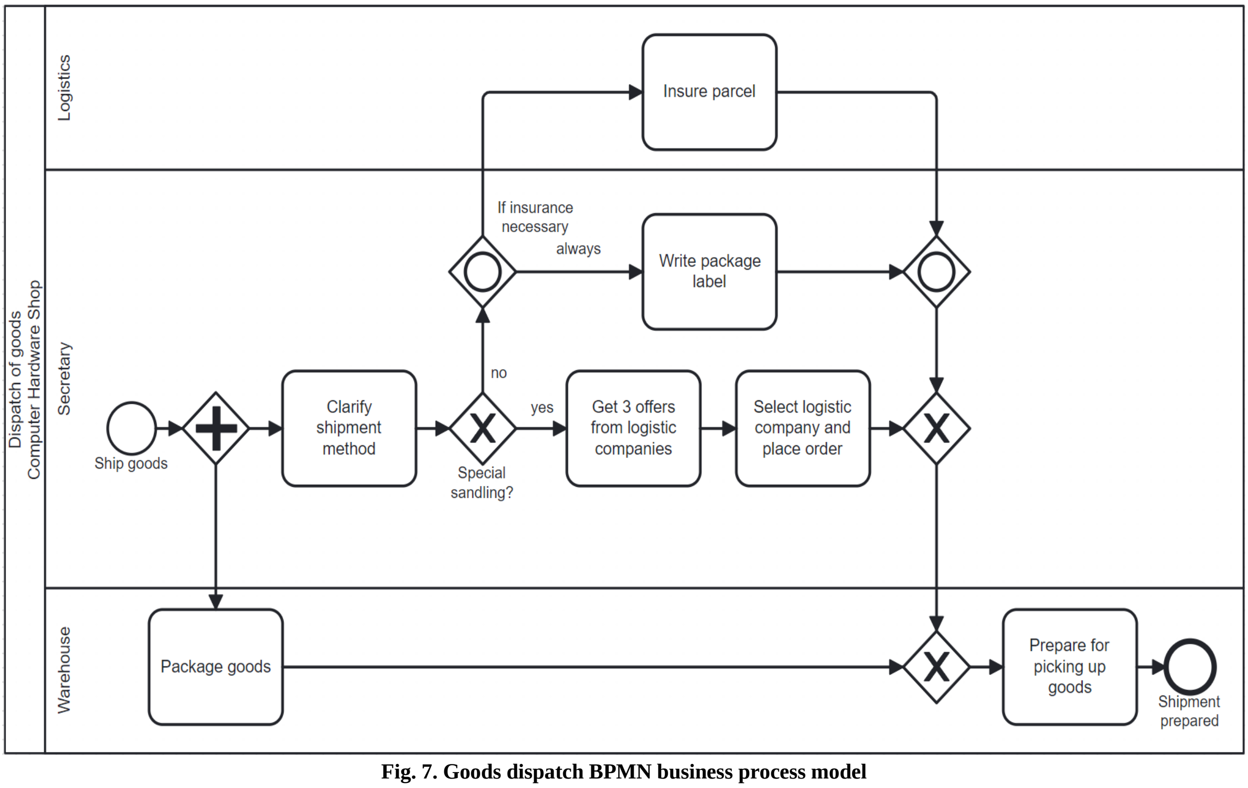

Fig. 7. Goods dispatch BPMN business process model

According to [16], this business process model represents the process described using the following text: "If goods shall be shipped, the secretary clarifies who will do the shipping. If you have large amounts, special shipping will be necessary. In these cases, the secretary invites three logistic companies to make offers and she selects one of them. In case of small amounts, normal post shipment is used. Therefore, a package label is written by the secretary and a parcel insurance taken by the logistics department head if necessary. In the meantime, the goods can be already packaged by the warehousemen. If everything is ready, the packaged goods are prepared for being picked up by the logistic company."

Let us use Python and NLTK (Natural Language Toolkit) [17] to implement the proposed approach and verify its efficiency. Input texts T_1 and T_2 are given in Fig. 8.

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

T2 = 'If goods shall be shipped, the secretary clarifies who will do the shipping. If you have large amounts, special shipping wi
print('\nT2:', T2)

bpmn_activities = ['Clarify shipment method', 'Package goods', 'Get 3 offers from logistic companies', 'Insure parcel', 'Write pa
print('\nBPMN activities:', bpmn_activities)
```

```
T2: If goods shall be shipped, the secretary clarifies who will do the shipping. If you have large amounts, special shipping wi
ll be necessary. In these cases the secretary invites three logistic companies to make offers and she selects one of them. In c
ase of small amounts, normal post shipment is used. Therefore a package label is written by the secretary and a parcel insuranc
e taken by the logistics department head if necessary. In the meantime the goods can be already packaged by the warehousemen. I
f everything is ready, the packaged goods are prepared for being picked up by the logistic company.

BPMN activities: ['Clarify shipment method', 'Package goods', 'Get 3 offers from logistic companies', 'Insure parcel', 'Write p
ackage label', 'Select logistic company and place order', 'Prepare for picking up goods']
```

```
T1 = ' '.join(bpmn_activities).strip()
print('\nT2:', T1)
```

```
T2: Clarify shipment method Package goods Get 3 offers from logistic companies Insure parcel Write package label Select logisti
c company and place order Prepare for picking up goods
```

Fig. 8. Input texts declared in the Python notebook

Fig. 9 demonstrates steps of the proposed algorithm (Fig. 3) sequentially applied to input texts T_1 and T_2, and obtained sets of words W_1'' and W_2'' as the result.

```
stop_words = set(stopwords.words('english'))
porter = PorterStemmer()
```

```
W1 = word_tokenize(T1)
W1 = [word.lower() for word in W1]
W1 = [word for word in W1 if word.isalpha()]
W1 = [word for word in W1 if not word in stop_words]
W1 = [porter.stem(word) for word in W1]
W1 = list(dict.fromkeys(W1))
print('\nW1:', W1)
```

```
W1: ['clarifi', 'shipment', 'method', 'packag', 'good', 'get', 'offer', 'logist', 'compani', 'insur', 'parcel', 'write', 'labe
l', 'select', 'place', 'order', 'prepar', 'pick']
```

```
W2 = word_tokenize(T2)
W2 = [word.lower() for word in W2]
W2 = [word for word in W2 if word.isalpha()]
W2 = [word for word in W2 if not word in stop_words]
W2 = [porter.stem(word) for word in W2]
W2 = list(dict.fromkeys(W2))
print('\nW2:', W2)
```

```
W2: ['good', 'shall', 'ship', 'secretari', 'clarifi', 'larg', 'amount', 'special', 'necessari', 'case', 'invit', 'three', 'logi
st', 'compani', 'make', 'offer', 'select', 'one', 'small', 'normal', 'post', 'shipment', 'use', 'therefor', 'packag', 'label',
'written', 'parcel', 'insur', 'taken', 'depart', 'head', 'meantim', 'alreadi', 'warehousemen', 'everyth', 'readi', 'prepar', 'p
ick']
```

Fig. 9. Results obtained using the proposed algorithm

The following figure (Fig. 10) demonstrates the computed value of the Jaccard coefficient [12] together with the cardinalities of sets of words W_1'' and W_2'', and their intersection $|W_1'' \cap W_2''|$.

```
card_W1 = len(W1)
print('\ncard(W1):', card_W1)

card_W2 = len(W2)
print('\ncard(W2):', card_W2)

card_W1_intersect_W2 = len(set(W1).intersection(set(W2)))
print('\ncard(W1 ∩ W2):', card_W1_intersect_W2)

jaccard = card_W1_intersect_W2 / (card_W1 + card_W2 - card_W1_intersect_W2)
print('\njaccard:', round(jaccard, 2))
```

```
card(W1): 18

card(W2): 39

card(W1 ∩ W2): 13

jaccard: 0.3
```

Fig. 10. Computed value of the Jaccard coefficient

Results given in Fig. 10 confirm that the similarity between the BPMN model representing goods dispatch business process (Fig. 7) and its textual description [16] is 0.3, according to the Jaccard coefficient [12].

Therefore, the compliance of the business process model (Fig. 7) with its text description can be estimated as 30%, which means that some information about the business process given in its description is missing in the BPMN model (i.e. $|W_1''| = 18 < |W_2''| = 39$).

The software solution for analyzing the compliance of business process models with their textual descriptions was developed using the .NET platform and the C# programming language [18]. It is proposed to choose a three-tier architecture (Fig. 11) to build the software for analyzing the compliance of business process models with their textual descriptions. The 3-tier architecture assumes that a software application consists of three components: a client application that interacts with an application server, which is connected to a database server (we use Microsoft SQL Server) [19].

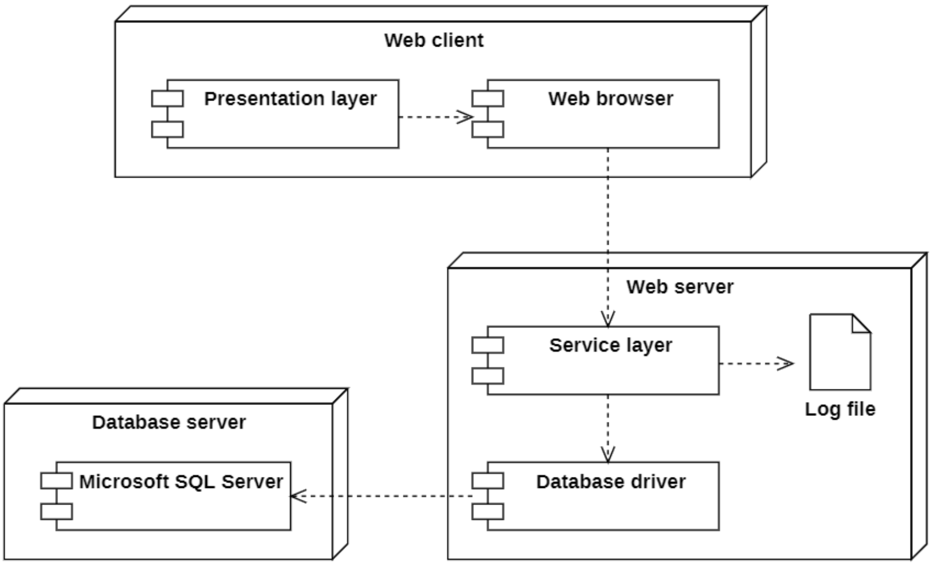

Fig. 11. Software architecture design

When the users open the application, they are taken to the main page of the website. The user should specify the name of a business process, upload a BPMN file and a text file describing the model. Then the user is redirected to the results page. It demonstrates the compliance of the business process model with its textual description, and the number of words in the result sets of words W_1'' and W_2'' are demonstrated on the results page as well (Fig. 12).

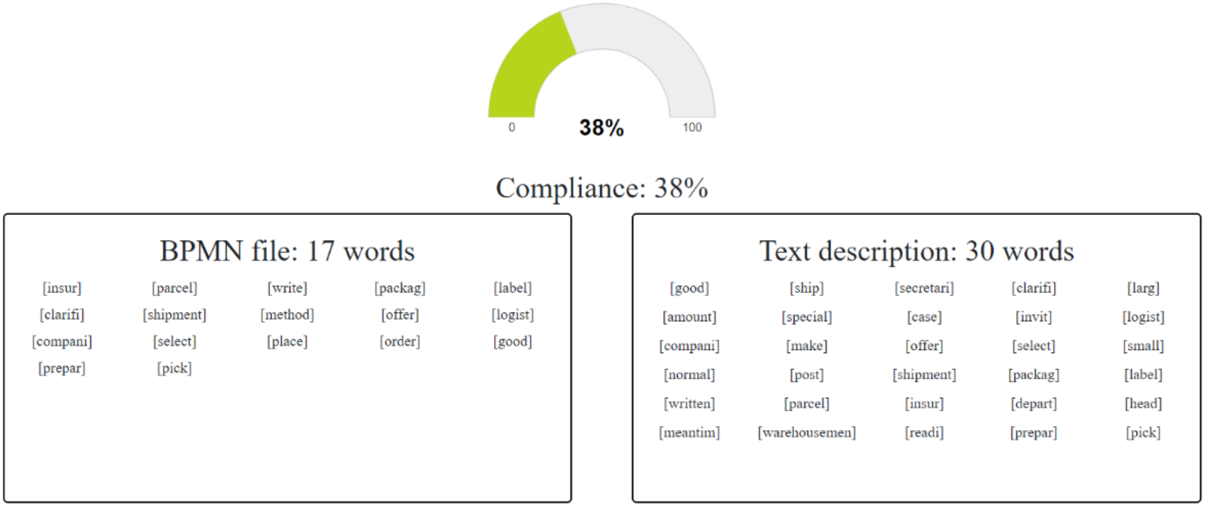

Fig. 12. Results page demonstrates obtained results

However, this approach has the limitation connected to the fact that rich business process descriptions may cause lower similarity values and, thus, signalize the false lower compliance of business process models. Hence, this limitation can be bypassed by the inclusion of other BPMN elements, such as events and gateways, into consideration when measuring the similarity index. Another limitation of the proposed approach is connected with the possible presence of synonyms and phrases that should not be divided into different words to keep their meaning.

Conclusions

In this research, the business process modeling is considered and the relevance of the problem of analysis of the compliance of business process models with their textual descriptions is defined. The following conclusions can be made:

1) the analysis and comparison of existing software tools for business process modeling and analysis have shown that widely-used and well-known tools do not allow checking the compliance of BPMN workflow diagrams with their textual descriptions;

2) an overview of NLP techniques, such as tokenization, removal of stop words, and stemming are discussed and considered as methods that can be used to solve the problem of analyzing the compliance of business process models with their textual descriptions;

3) the proposed algorithm based on NLP techniques solves the formulated problem of the analysis of business process models compliance with respective textual descriptions and produces estimations in the range between 0 and 1 for the very bad and very good compliance respectively.

In the future, it is necessary to elaborate the software that will help business users analyze the compliance of BPMN models with textual descriptions of depicted workflows, as well as to elaborate the developed algorithm by using advanced artificial intelligence methods, e.g. neural networks, trained on the collection of real business process models.

References

1. Suchenia A. Overview of verification tools for business process models / A. Suchenia, P. Wiśniewski, A. Ligęza // Annals of Computer Science and Information Systems. – 2017. – Vol. 13. – P. 295-302.
2. Allweyer T. BPMN 2.0: introduction to the standard for business process modeling / T. Allweyer // BoD – Books on Demand. – 2016. – 172 p.
3. Be Efficient: Bonitasoft Introduces New Bonita BPM 6 Platform, https://www.businesswire.com/news/home/20130605006087/en/Efficient-Bonitasoft-Introduces-Bonita-BPM-6-Platform, 30.05.2023.
4. SAP Signavio, https://www.signavio.com/, 30.05.2023.
5. ProcessMaker Review, https://comparecamp.com/processmaker-review-pricing-pros-cons-features/, 30.05.2023.
6. Process Documentation – Bizagi Models, https://help.bizagi.com/process-modeler/en/index.html?where_to_share2.htm, 30.05.2023.
7. Tokenization in NLP: Types, Challenges, Examples, Tools, https://neptune.ai/blog/tokenization-in-nlp, 30.05.2023.
8. Tokenization in NLP, https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/, 30.05.2023.
9. Stop Words in NLP, https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47, 30.05.2023.
10. Jongejan B. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike / B. Jongejan, H. Dalianis // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. – 2009. – P. 145-153.
11. Business Process Model and Notation (BPMN), Version 2.0, https://www.omg.org/spec/BPMN/2.0/PDF, 30.05.2023.
12. Warrens M. J. Similarity coefficients for binary data: properties of coefficients, coefficient matrices, multi-way metrics and multivariate coefficients / M. J. Warrens // Leiden University. – 2008. – 254 p.

13. Rudskyi O. V. Towards the analysis of the compliance of business process models with their textual descriptions / O. V. Rudskyi, A. M. Kopp, D. L. Orlovskyi // Information technologies: science, technology, technology, education, health: abstracts of reports of XX international scientific and practical conference MicroCAD-2022. – 2022. – P. 822.
14. Tielietov O. S. Four-vector efficiency of infrastructure in the system of providing regional socially significant needs taking into account the concept of marketing of changes / O. S. Tielietov, N. Y. Letunovska, Y. M. Melnyk // Bioscience Biotechnology Research Communications. – 2019. – Vol. 12, N. 3. – P. 637-645.
15. Rudskyi O. V. Software requirements elicitation for the analysis of the compliance of business process models with their textual descriptions / O. V. Rudskyi, A. M. Kopp // Materials of the 5th All-Ukrainian scientific and practical Internet conference of students, postgraduates and young scientists on the topic "Modern computer systems and networks in management". – 2022. – P. 158-161.
16. BPMN for research, https://github.com/camunda/bpmn-for-research, 30.05.2023.
17. NLTK :: Natural Language Toolkit, https://www.nltk.org/, 30.05.2023.
18. A tour of the C# language // URL: https://learn.microsoft.com/uk-ua/dotnet/csharp/tour-of-csharp/, 30.05.2023.
19. Fowler M. Patterns of Enterprise Application Architecture / M. Fowler // CreateSpace Independent Publishing Platform. – 2017 – 560 p.

Oleksandr Rudskyi **Олександр Рудський**	Master's Student of Software Engineering and Management Intelligent Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: oleksandr.rudskyi@cs.khpi.edu.ua. https://orcid.org/0009-0001-1130-9957	магістрант кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна
Andrii Kopp **Андрій Копп**	PhD, Associate Professor, Associate Professor of Software Engineering and Management Intelligent Technologies Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: andrii.kopp@khpi.edu.ua https://orcid.org/0000-0002-3189-5623	доктор філософії, доцент, доцент кафедри програмної інженерії та інтелектуальних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна