Natural Language Processing for Risk Identification in Business Process Repositories

Avi Wasser[1] and Maya Lincoln[2]
[1]*University of Haifa, Mount Carmel, Haifa, Israel*
[2]*ProcessGene Ltd, MATAM Science Park, Haifa, Israel*

Keywords: Risk Management, Risk Design, Business Process Repositories, Business Process Management, Natural Language Processing (NLP), ProcessGene.

Abstract: In recent years, researchers have become increasingly interested in developing methods and tools for automating the design of governance, risk and compliance (GRC) models. This work suggests a method for machine-assisted identification and design of new risks, based on business logic that is extracted from real-life process repositories using a linguistic analysis of the operational similarity between process conducts. The suggested method can assist process analysts, audit executives and risk managers in identifying new organizational risks while making use of knowledge that is encoded in existing process repositories. The suggested framework was tested on the ProcessGene process repository, showing our approach to be effective in enabling the identification and design of new risks within real-life business process models.

1 INTRODUCTION

With the increase of regulatory requirements on one hand, and the attempts to optimize business outcomes on the other, organizations are required to invest more efforts in identifying, managing and mitigating risks. Executive officers are specifically required to demonstrate effective risk management practices, and to ensure corporate transparency and visibility into the business. The risk management process is continuous, and needs to be closely monitored. As management is personally responsible for monitoring risk levels, this responsibility requires significant management attention and allocation of time and effort.

Risk modelling is considered a manual, labour intensive task, whose outcome depends on personal domain expertise with errors or inconsistencies that lead to bad risk prevention and high risk related costs (Muller et al, 2007). Hence, automating the identification and design of risks does not only save design time but also supports non-expert designers in defining new risks within business process models.

While some works focus on the design of new process models, mostly for a specific functional domain, none refers to new risk definition and identification within process repositories.

This work aims to suggest a generic method for designing risks within business process models related to any functional domain. The suggested method guides business analysts and risk managers that opt to identify and design new risks, by suggesting new risks within a process repository. The business logic for such suggestions is extracted from process repositories through the analysis of existing business process activities and their related risks. We show through an empirical evaluation that by utilizing operational process similarity analysis, it is possible to effectively support the design of new risks within process models.

The work proposes an innovative method for assisting designers in designing brand new risks while making use of knowledge that is encoded in the design of existing, related process models. Our work presents the following innovations: (a) it provides generic support to the design of new risks within existing process repositories; (b) it extends the PDC model (Lincoln et al, 2007) to support the representation of risks; and (c) it enables the design of new risks based on extraction of business logic from business process repositories.

The rest of the paper is organized as follows: we present related work in Section 2, positioning our

Wasser, A. and Lincoln, M.
Natural Language Processing for Risk Identification in Business Process Repositories.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 608-613
ISBN: 978-989-758-187-8

work with respect to previous research. In Section 3 we present an extended model for representing process risks based on the process descriptor notion, presented first in the work of Lincoln et al (2007), and extended in this work for the field of new risk design. Then, we describe our method for designing new risks in Section 4. Section 5 introduces our empirical analysis. We conclude in Section 6.

2 RELATED WORK

Most of the efforts invested in developing methods and tools for designing process models focus on supporting the design of alternative process steps within existing process models. Such a method is presented by Schonenberg et al (2008) aiming to provide next-activity suggestions during execution based on historical executions and optimization goals. Similarly, Gschwind et al (2008) suggest an approach for helping business users in understanding the context and consequences of applying pre-defined patterns during a new process design.

Few works were devoted to the design of brand new process models within specific and predefined domains. The work presented by Muller et al (2007) utilizes the information about a product and its structure for modeling large process structures. Reijers et al (2003) present a method, for designing new manufacturing related processes based on product specification and required design criteria. Works in the domain of risk design also focus on specific risk domains, such as credit risks (Giesecke, 2004; Galindo and Tamayo, 2000), inventory management risks (Michalski, 2009), and financial risks (Barbaro and Bagajewicz, 2004). Our work offers a generic design method that is domain agnostic and does not rely on product design data. In addition our work assists in the design of risks rather than process activities.

A requirement for the support of business process design involves the performance of a structured reuse of existing building blocks and pre-defined patterns that provide context and sequences (Gschwind et al, 2008). The identification and choice of relevant process components are widely based on the analysis of linguistic components - actions and objects that describe business activities. Most existing languages for business process modeling and implementation are activity-centric, representing processes as a set of activities connected by control-flow elements indicating the order of activity execution (Wahler and Kuster, 2008). In recent years, an alternative approach has been proposed, which is based on objects (or artifacts/entities/documents) as a central component for business process modeling and implementation. Our work supports this approach and focuses on objects for the purpose of risk identification and modeling.

Finally, the work of Lincoln et al (2007) presents the concept of business process descriptor that decomposes process names into objects, actions and qualifiers. In this work we take this model a significant step forward by extending the framework to support also the representation of risks using a new taxonomy - the "risk descriptor."

3 THE DESCRIPTOR MODEL

In the Process Descriptor Catalog model ("PDC") (Lincoln et al, 2007) each activity is composed of one action, one object that the action acts upon, and possibly one or more action and object qualifiers, as illustrated in Figure 1, using UML relationship symbols. Qualifiers provide an additional description to actions and objects. In particular, a qualifier of an object is roughly related to an object state. State-of the art Natural Language Processing (NLP) systems, e.g., the "Stanford Parser" (Stanford parser, 2016), can be used to automatically decompose process and activity names into process/activity descriptors.

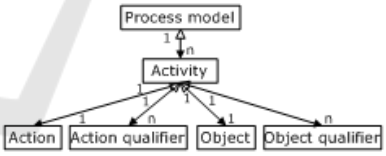

Figure 1: The activity decomposition model.

For example, the activity "Manually Calibrate the Color Machine" generates an activity descriptor containing the action "calibrate, the action qualifier "manually, the object "machine and the object qualifier "color." In short, this descriptor can be represented as the tuple (calibrate,manually,machine,color) - where the action and its qualifier are followed by the object and its qualifier. In general, such tuple can be represented as (A,AQ,O,OQ), where A represents the action, AQ represents the action qualifier, O represents the object and OQ represents the object qualifier.

In this work we extend the descriptor model for representing risk names. To do that, we have analysed 842 real-life risk names from the ProcessGene repository (ProcessGene, 2016), e.g. "employee has criminal records", "corrupt backup tapes", "signature forgery". We found out that in 98.2% of the cases risks are linguistically phrased as states (an object with 0-n qualifiers) without any action and action qualifiers. We also found out that it was possible to convert the other 1.8% of the risk names into a state format (e.g. the risk "Data Restoration will Fail" can be converted into the state: "Data Restoration Failure"). Therefore, risk names can be represented by a partial (degenerate) descriptor model, with null values for the action related constructs. For example, the risk "Backup Restoration Failure" can be represented by a descriptor containing a NULL action, NULL action qualifiers, the object "backup and the object qualifier "restoration failure. In short, the risk's name can be represented by the tuple (NULL, NULL, backup, restoration failure).

4 THE RISK IDENTIFIER METHOD

The Risk Identifier method analyses an underlying process repository and suggests the addition of missing risks.

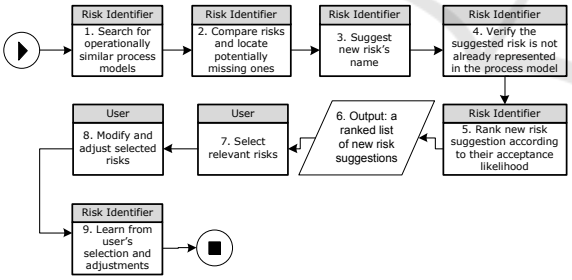

Figure 2: The risk identifier mechanism.

The risk identifier is illustrated in Figure 2. The identification process starts when a user (e.g. a risk manager or a process designer) seeks to reveal additional risks that are not represented within a process repository. In response, the risk identifier searches the process repository and retrieves operationally similar processes (phase #1, see elaboration in Section 4.1). It then compares the risks at each process model and identifies potentially missing ones - meaning risks that exist in one process model (namely "reference risks") and are missing at a similar process model (namely, an "examined process model") (phase #2). Based on the examined process model content on one hand, and the reference risk name on the other hand, the risk identifier suggests a name for the new risk candidate (phase #3, elaborated in Section 4.2).

Going back to the examined process model, and prior to suggesting the new risk to the user, the risk identifier verifies that the new risk candidate is not already represented in the examined process model (phase #4, Section 4.3). The new risk candidates are then ranked according to their user acceptance likelihood. This is done using a learning mechanism that collects data regarding the user inputs in previous risk identification procedures (phase #5, Section 4.4). This results in an output of a ranked list of new risk candidates that is presented to the user (phase #6).

The user reviews the output option list and selects the relevant risks that in her opinion should be added to the process repository (phase #7). She then modifies and adjusts the auto-generated risk names (phase #8, Section 4.5). The risk identifier analyses the user's selection and modifications and adjusts his underlying learning mechanism for future risk identification runs (phase #9).

4.1 Searching for Operationally Similar Process Models

The goal of this phase is to find process models within the process repository that are similar in operation terms. Such processes achieve a different business goal but are similar in the way (how) they are conducted. Since risks refer to the modus operandi facet of the process, it is assumed that such similar processes will also share similar risks. This assumption is validated and supported by experiments in Section 5.

To perform such operation-based search we apply the search method presented by Lincoln and Gal (2011). As an example, the following process models were found as operationally similar: "Background check-up of a new employee candidate" and "supplier assessment." Although these process models are not semantically similar, their activity flow is similar. Note that a semantic-based similarity search would not have found the two example processes since they do not share similar terminology but are rather similar in the way they are executed. For a more profound discussion

on operational vs. semantic process search see the work of Lincoln and Gal (2011).

4.2 Suggesting a Name for the New Risk Candidate

The goal of this phase is to construct an appropriate name for the new risk ("nr") candidate. To do this the following procedure is detailed as follows.

1. Decompose the reference risk ("rr") name into a risk descriptor (see Section 3), represented as ($NULL,NULL,O_{rr},OQ_{rr}$).

2. Decompose the reference process ("rp") name into a process descriptor (see Section 3), represented as ($A_{rp},AQ_{rp},O_{rp},OQ_{rp}$).

3. Decompose the examined process ("ep") name into a process descriptor (see Section 3), represented as ($A_{ep},A_{ep},O_{ep},OQ_{ep}$).

4. If $O_{rr}=O_{rp}$ then the new risk candidate name is: ($NULL,NULL,O_{ep},OQ_{rr}$).

5. Else: the new risk candidate name is: ($NULL,NULL,O_{rr},OQ_{rr}$).

Following the above example, the risk "Missing data about the employee candidate" was located within the process model of "Background check-up of a new employee candidate." As a consequence, according to this phase, a new risk candidate was offered to the "Supplier assessment" process, named: "Missing data about the supplier."

4.3 Redundancy Check-up

This phase aims at verifying that the new risk candidate is not already represented as part of the examined process model. To do that, it is required to semantically compare between each of the risks related to the examined process and the new risk candidate's name.

4.4 Preparing a Set of Output Options

The input for this phase is a set of several new risk candidates. In order to assist the user in reviewing the list options, this phase aims at ranking the new risk candidate list according to their user acceptance likelihood. This is done by learning and analysing the user's inputs in previous risk identification procedures as follows.

At the end of each risk identification procedure the user is required to select relevant risks and also modify and adjust selected risks (phases #7 and #8 in Figure 2). These user actions from previous runs influence the grade each new risk candidate receives at the current run in the following way. First, each risk candidate is labeled with an identical grade=1. Then, the grade of each risk candidate is adjusted as follows:

1. *Penalty Due to Risk Content.* This correction aims to add a penalty to risks that are usually removed, by increasing the risk candidate's grade according to the number of times a risk with the same name was marked as irrelevant in previous runs. Similarly, the grade increases according to the number of times a risk with the same name was added to the repository in previous runs. Such additions of the same risk can also indicate that this risk was neglected in the original process repository.

2. *Penalty Due to the Nature of the Examined Process.* This correction aims to take into account the "safety" nature of the examined process. A "safe" process is a process that the chance of adding risks to it are low (it may consist risks but they are already being fully handled by the process). Therefore, the risk's grade is increased according to the number of times a risk within the same examined process model was marked as irrelevant in previous runs. Similarly, the risk's grade is decreased according to the number of times a risk within the same examined process model was added in previous runs.

3. *Penalty Due to Inaccuracy.* This correction aims to take into account the expected accuracy of the risk. Therefore, the grade will increase according to the total number of changes the user made to the descriptor of the same risk name in previous runs. A change is considered as any replacement of an object or an object qualifier name.

After calculating each of the new risk list options, the risk identifier sorts the list in an ascending order - from the most probable to the most improbable in terms of the chances the user will finally accept and add the risk to the process repository.

4.5 Adjusting the Auto-generated Risk Names

In case the user decides that the newly suggested risk is relevant to the examined process, she re-examines its name and optionally modifies one or more of its risk descriptor components (object and/or object qualifiers).

5 EXPERIMENTS

We now present an empirical evaluation of the proposed method effectiveness. First, we present the experimental setup and describe the data sets that were used. Based on this setup we present the implemented methodology. Finally, we present the experiment results and provide an empirical analysis of these results.

5.1 Data

We chose a set of 43 process models from the ProcessGene repository that have at least one operationally similar process. The selected process models are part of different business categories (e.g. manufacturing, procurement, human resource management) and have a different number of risks.

5.2 Evaluation Methodology

To evaluate the suggested method we conducted 43 experiments. At each experiment, the risks of a single process were removed from the database and then reconstructed using the risk identifier method. This "machine assisted reconstruction" enables us to objectively measure the method's effectiveness. In addition, a risk management expert was asked to assess risks that were offered by the risk identifier and did not belong originally to the process repository.

5.3 Results and Analysis

Table 1 presents a summary of the experiment results. On average, 24% risks from the ProcessGene repository were missing from the generated risk lists (see column #1). This means that 76% of the risks in the repository were reconstructed successfully by the risk identifier mechanism, showing a high level of usefulness to the method.

Table 1: Experiment results.

Column #	1	2	3
Column name	% of missing risks in the generated risk list	% of redundant risks in the generated risk list	% of generated risks that are not represented in the ProcessGene repository
Avg.	24%	9%	4%

In addition, on average, 9% of the risks candidates were redundant - meaning they were not relevant to the examined process (see column #2), again, highlighting the level of accuracy of the generated risk list.

Finally, on average, the risk management expert chose to add 4% of the risks that were not represented in the ProcessGene repository (see column #3). This indicates that the risk identifier mechanism's ability to generate new risks was not dependent on the specific given repository.

To summarize, the experiments have demonstrated the usefulness of the machine-based risk identification assistant in constructing risks to process models. We have also measured and evaluated the effectiveness of the method in the given experimental setup, both in terms of the amount of missing risks and in the amount of redundant risks.

6 CONCLUSIONS

We proposed a mechanism for automating the generation of risks within a process repository. Such a mechanism saves design time and supports non-expert designers in creating new risks for business process models. The proposed method and experiments provide a starting point that can already be applied in real-life scenarios, yet several research issues remain open, including: (1) an extended empirical study to further examine the quality of the framework; (2) extending the method to include mitigating controls as well; and (3) extending the learning mechanism to further predict the user's behaviour.

REFERENCES

Barbaro, A., Bagajewicz, M., J., 2004. Managing Financial risk in planning under uncertainty. *AIChE Journal, 50(5):963989.*

Galindo, J., Tamayo, P., 2000. Credit risk assessment using statistical and machine learning: basic methodology and risk modelling applications. *Computational Economics, 15(1-2):107143.*

Giesecke, K., 2004. Credit risk modelling and valuation: An introduction. Available at *SSRN 479323.*

Gschwind, T., Koehler, J., Wong, J., 2008. Applying patterns during business process modelling. In *BPM, volume 5240, pages 419. Springer.*

Lincoln, M., Gal, A., 2011. Searching business process repositories using operational similarity. On the Move

to Meaningful Internet Systems: *OTM 2011, pages 219.*

Lincoln, M., Karni, R., Wasser, A., 2007. A Framework for Ontological Standardization of Business Process Content. *International Conference on Enterprise Information Systems, pages 257263.*

Michalski, G., 2009. Inventory management optimization as part of operational risk management. *Economic Computation and Economic Cybernetics Studies and Research, pages 213222.*

Muller, D., Reichert, M., Herbst, J., 2007. Data-driven modelling and coordination of large process structures. Lecture Notes in *Computer Science, 4803:131.*

ProcessGene website, 2016. http://processgene.com/business-process-repository/.

Reijers, H.A., Limam, S., Van Der Aalst, W.M.P., 2003. Product-based workflow design. *Journal of Management Information Systems, 20(1):229262.*

Schonenberg, H., Weber,B., van Dongen, B.F, van der Aalst, W.M.P., 2008. Supporting flexible processes through recommendations based on history. In *International Conference on Business Process Management (BPM 2008), volume 5240, pages 5166. Springer.*

Stanford parser, 2016. http://nlp.stanford.edu:8080/parser/index.jsp.

Wahler, K., Kuster, J.M., 2008. Predicting Coupling of Object-Centric Business Process Implementations. In *Proceedings of the 6th International Conference on Business Process Management, page 163. Springer.*