Master Thesis Proposal

Kefang Ding

August 15, 2018

Title:	Process Enhancement by Incorporating Negative Information in Model Repair
Author:	Kefang Ding
Supervisor:	Professor Dr. ir. Wil van der Aalst Dr. ir. Sebastiaan J. van Zelst
Degree:	Master of Science

Introduction

Process mining techniques aim to support the analysis of business processes based on event logs. These techniques focus on three tasks: process discovery, conformance checking and model enhancement. Process discovery aims to discover a new model from an event log in order to understand the business process execution. Conformance checking analyzes the deviation between process models and observed behavior during execution. Enhancement improves an existing model based on an event log by extending the model with more data perspectives or repairing the existing model to better reflect observed behaviors.

Between those techniques, repair model belongs to process enhancement and stands between process discovery and conformance checking. It improves the existing model by adding loops, subprocess, and removing infrequent transitions [1]. Later in [2] [3], focuses are not only on fitness improvement but also on good KPI outcomes. However, the state-of-the-art model repair methods only consider the positive information which benefits the good performance, but leave out the impact from negative information. It usually results in a less precise model, which can be shown in the following example.

The original model is defined in the Figure 1 (a), where A is followed directly by B. During its execution in real life, an event log is generated:

$$< A, B >^{55}, < B, A >^{105}$$

After current model repair techniques in [3][2], a new process model with high fitness is generated. A and B now have parallel relation. If we analyze the impact on performance according to KPI, the event log is divided into positive and negative set:

$$Positive\ examples : < A, B >^5, < B, A >^{100}$$

$$Negative\ examples : < A, B >^{50}, < B, A >^5$$

Under consideration of only positive information, the repaired model keeps the same like in Figure 1(b), since both $< A, B >$ and $< B, A >$ contributes to good performance.

(a) original process model

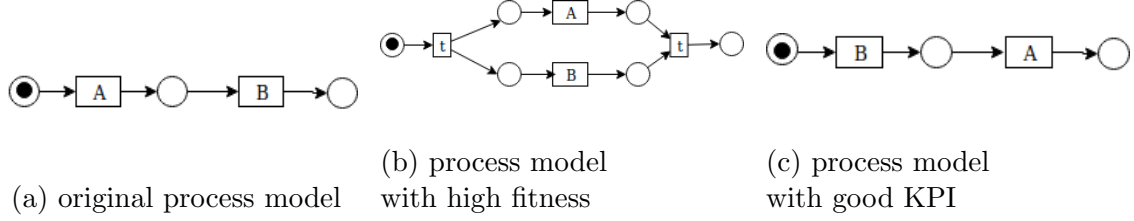(b) process model with high fitness

(c) process model with good KPI

Figure 1: example for model change under model repair

However, it's obvious that $< A, B >$ leads to more bad performance than good ones and should be avoided. The Figure 1(c) shows the expected model with incorporating the negative information. This model reinforces positive examples and avoids negative examples, which provides us a better business execution process.

On the other side, the current methods in [3][2] are not capable to combine multiple KPI at the same time, which leads the generation of good model difficult.

Aim

Given input of one existing process model, an event log and KPIs, our aim is to improve current process enhancement techniques by incorporating negative information within process model repair, and generate a better process model with higher fitness, precision and better KPIs outcomes. Therefore, the repaired model provides a better way to understand and execute the real business process.

To evaluate the improvement, we will build a confusion matrix in the Table 1. The columns represent model allowed behavior and not allowed behavior; the rows means the positive examples and negative ones in the event log. Good improvement should increase the diagonal ratio in the table, namely the AP and NN, while lowering the percentage of reverse diagonal, which is PN and AN.

Table 1: confusion matrix for evaluation

	Allowed behavior	Not allowed behavior
Positive Examples	AP	PN
Negative Examples	AN	NN

Under the current techniques for repair model, only positive example are considered and the model in the middle is generated. In the view of its confusion matrix in Table 1, it has less precision by allowing all behavior in model.

Table 2: confusion matrix for given example in current approach

	Allowed behavior	Not allowed behavior
Positive Examples	105	0
Negative Examples	55	0

In our approach, both negative and positive examples are used to generate expected model. The confusion matrix is driven in Table 3.

Table 3: confusion matrix for given example in improved approach

	Allowed behavior	Not allowed behavior
Positive Examples	100	5
Negative Examples	5	50

Also, given multiple KPIs, it should be able to combine them together and provide a balanced model with good performance. But this suggestion should be based on the hypothesis:

Method

The inputs for process model enhancement within model repair includes the following data:

- Existing process modeled using Petri net

- Event Log corresponding the actual execution of modeled process

- Key Performance Indicators(KPIs)

Our approach produces a repaired model as output and its workflow is shown in the Figure 2. The first phase in shadow accepts an existing process and an event log as input, then repairs model w.r.t. fitness. After that the event log is classified into positive and negative examples w.r.t. KPIs, both positive and negative examples are passed to the second phase besides the repaired model with good fitness. At last, the model is modified to reinforce the good examples while avoiding the negative examples in event log.

The first phase is optional, if the process model has good fitness, it is passed directly to the phase to repair model w.r.t. KPI. If not, we use the model repair framework in [1]. Subprocess and loops are inserted into the original model. Infrequent paths are removed later to keep the model with good fitness and better precision.
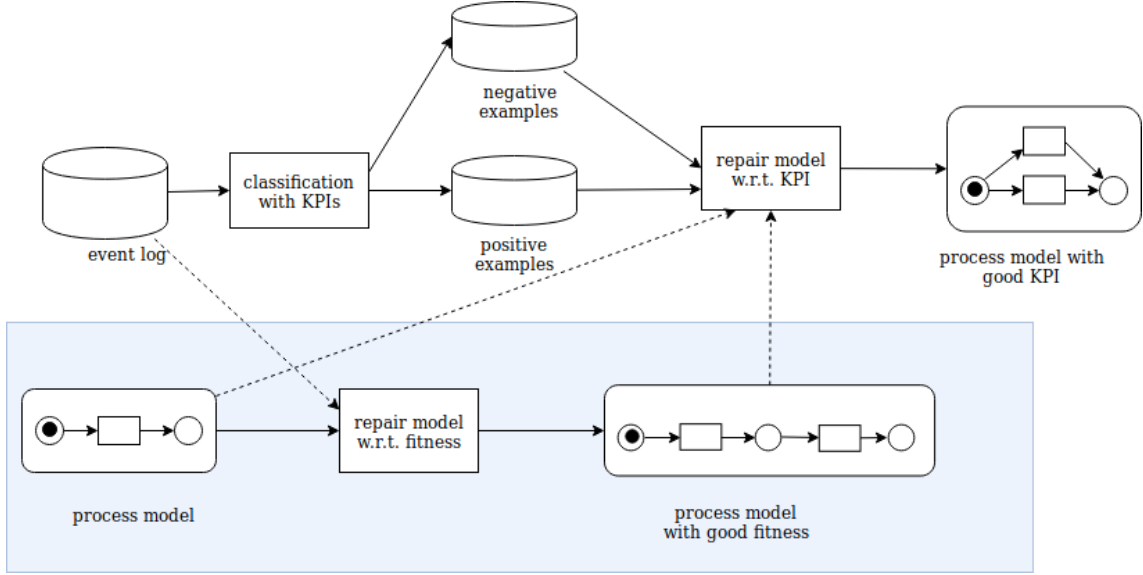
Figure 2: our approach to repair model

To classify the event log with KPIs, decision tree, associated rules are used.Later, another improvement of the approach can be introduced to combine multiple KPIs, and quantify the goodness of event trace. However, it is based on the following hypothesis.

- By implying the quantitative impact from multiple KPIs on model repair, we could get better result than by only using the classification method.

Now, to incorporate negative information on model improvement, we are motivated from [7] [4] [6] [9] which uses Inductive Logic Programming techniques like TILDE or SCIFF to get model constraints from positive and negative examples. Based on this, we propose our own method to repair model.

- change Petri net into rules constraints.
  Business Process in Petri net imposes the (minimal) set of constraints that must be satisfied when executing the process activities. Thoese constraints can be expressed into a declarative language like ConDec.

- train rules based on the positive and negative examples and get modified rules.
  After the transformation from Petri net, we have the set of existing constraints which enable both the positive and negative performance. By using ILP techniques, we can find a set of rules under the current existing constraints, which separates the given positive and negative examples.

- transform the modified rules again into Petri net
  Improved Constraints on business process are used to generate a new Petri net.

To make sure the method above feasible, several hypotheses are needed:

1. The mapping from Petri net patterns and the rules in first-order logic is possible and bijective, such that the two-sided transformation is feasible.

2. Generated model considers the impact from positive and negative examples but still keeps the similarity to the original model.

The next primary work should test on the hypotheses.

Software and Hardware Requirements

The platform is ProM6, a open source project for process mining. To implement our approach, those plugins are in demand.

- repair w.r.t. fitness

    Repair Model in Uma package

- repair w.r.t. KPI

    Inductive Logic Programming

    Decision Tree : weka

    Associated Rules : weka

    Regression function : weka

Timeline

The original plan for our project is in the table listed below.

Table 4: time line for the work

time	taks
August	Test the proposed hypotheses by creating small demo
September	Register master thesis and design the architecture
October	Coding and debug
November	Test and Write Thesis
December	Write thesis and representation

References

[1] Fahland, Dirk, and Wil MP van der Aalst. "Model repairaligning process models to reality." Information Systems 47 (2015): 220-243.

[2] Dees, Marcus, Massimiliano de Leoni, and Felix Mannhardt. "Enhancing process models to improve business performance: a methodology and case studies." OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Cham, 2017.

[3] Adriansyah, Arya, et al. "Alignment based precision checking." International Conference on Business Process Management. Springer, Berlin, Heidelberg, 2012.

[4] vanden Broucke, Seppe KLM, et al. "Determining process model precision and generalization with weighted artificial negative events." IEEE Transactions on Knowledge & Data Engineering 1 (2013): 1.

[5] Buijs, Joos CAM, et al. "Improving business process models using observed behavior." International Symposium on Data-Driven Process Discovery and Analysis. Springer, Berlin, Heidelberg, 2012.

[6] Montali, Marco, et al. "Declarative specification and verification of service choreographiess." ACM Transactions on the Web (TWEB) 4.1 (2010): 3.

[7] Pesic, Maja, and Wil MP Van der Aalst. "A declarative approach for flexible business processes management." International conference on business process management. Springer, Berlin, Heidelberg, 2006.

[8] Lamma, Evelina, et al. "Applying inductive logic programming to process mining." International Conference on Inductive Logic Programming. Springer, Berlin, Heidelberg, 2007.

[9] Chesani, Federico, et al. "Exploiting inductive logic programming techniques for declarative process mining." Transactions on Petri Nets and Other Models of Concurrency II. Springer, Berlin, Heidelberg, 2009. 278-295.