Information Systems 47 (2015) 220–243

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Model repair — aligning process models to reality

Dirk Fahland*, Wil M.P. van der Aalst

Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Available online 27 December 2013

Keywords:
Process mining
Model repair
Petri nets
Conformance checking

ABSTRACT

Process mining techniques relate observed behavior (i.e., event logs) to modeled behavior (e.g., a BPMN model or a Petri net). Process models can be discovered from event logs and conformance checking techniques can be used to detect and diagnose differences between observed and modeled behavior. Existing process mining techniques can only uncover these differences, but the actual repair of the model is left to the user and is not supported. In this paper we investigate the problem of *repairing a process model w.r.t. a log* such that the resulting model can replay the log (i.e., conforms to it) and is as similar as possible to the original model. To solve the problem, we use an existing conformance checker that aligns the runs of the given process model to the traces in the log. Based on this information, we decompose the log into several sublogs of non-fitting subtraces. For each sublog, either a loop is discovered that can replay the sublog or a subprocess is derived that is then added to the original model at the appropriate location. The approach is implemented in the process mining toolkit ProM and has been validated on logs and models from several Dutch municipalities.

1. Introduction

Process mining techniques aim to extract non-trivial and useful information from event logs [1,2]. The process mining spectrum ranges from operational support techniques (predictions and recommendations) to techniques that identify bottlenecks and decision rules [1]. The two main (and best known) types of process mining are (1) process discovery and (2) conformance checking.

Process discovery techniques automatically construct a process model (e.g., a Petri net or a BPMN model) from an event log [1,3–8]. The basic idea of control-flow discovery is very simple: given an event log containing a collection of traces, automatically construct a suitable process model "describing the behavior" seen in the log. However, given the characteristics of real-life event logs, it is notoriously difficult to learn useful process models from such logs. Event logs only contain example behavior and do not explicitly indicate what is impossible. The fact that an event log does not contain a particular trace does not imply that that trace is impossible. Moreover, a process discovery technique needs to mediate between different concerns, e.g., *fitness* (ability to explain observed behavior), *simplicity* (Occam's Razor), *precision* (avoiding underfitting), and *generalization* (avoiding overfitting).

The second type of process mining is *conformance checking* [4,9–17]. Here, an existing process model is compared with an event log of the same process. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. The conformance check could yield that the model *does not describe the process executions observed in reality*: activities in the model are skipped in the log, the log contains events not described by the model, or activities are executed in a different order than described by the model.

In case an existing process model does not conform to reality one could — in principle — use process discovery to obtain a model that does. However, the discovered model is likely to bear no similarity with the original model, discarding any value the original model had, in particular if the original was created manually. A typical real-life example is the reference process model of a Dutch municipality as shown in Fig. 1(left); when rediscovering the actual process using logs from the municipality one would obtain the model in Fig. 1(right).

* Corresponding author.
E-mail addresses: d.fahland@tue.nl (D. Fahland), w.m.p.v.d.aalst@tue.nl (W.M.P. van der Aalst).

0306-4379/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.12.007

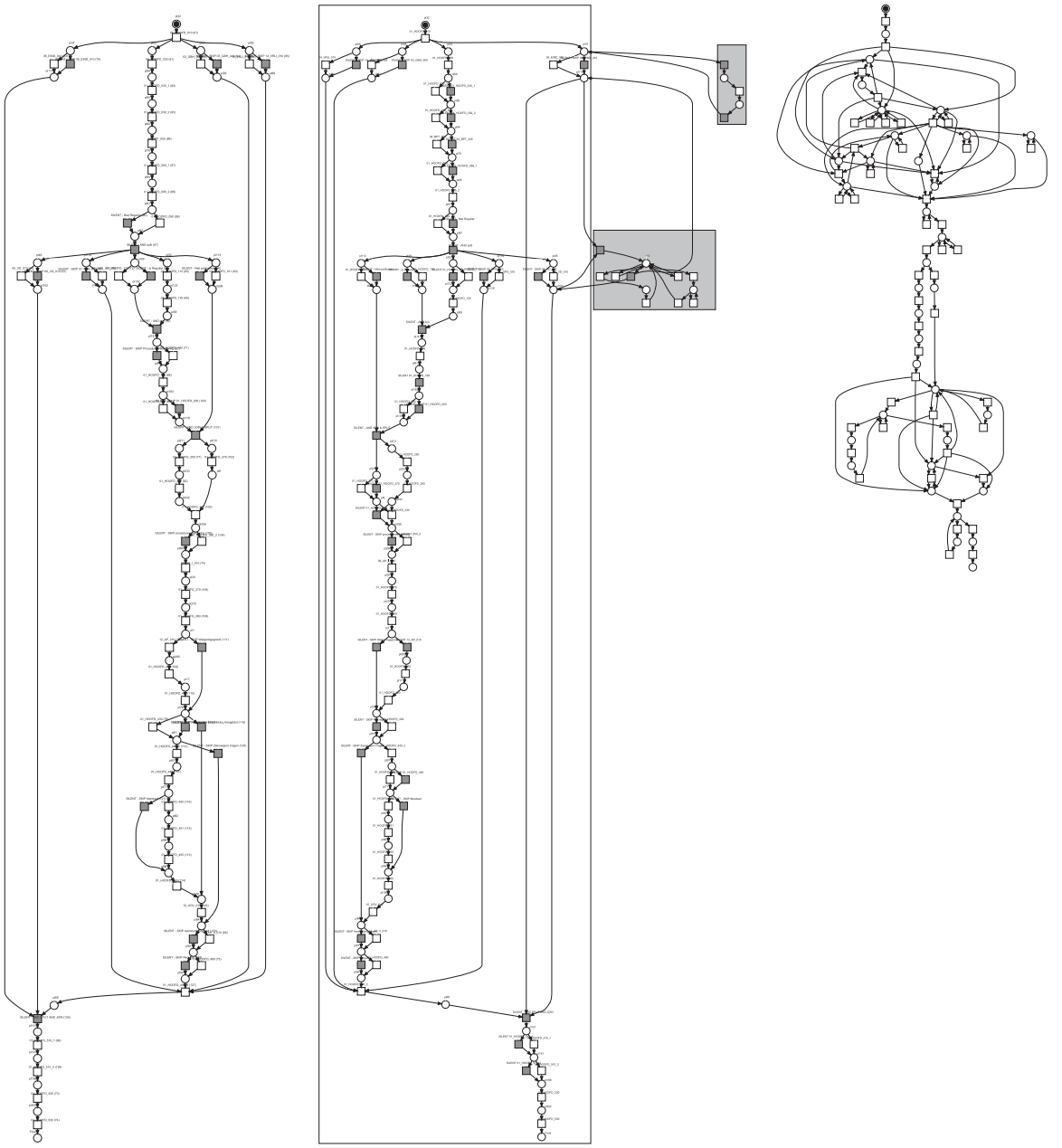

Fig. 1. Original model (left), model (middle) obtained by repairing the original model w.r.t. a given log, and model (right) obtained by rediscovering the process without considering the original model. The highlighted parts in the repaired model have been added to better fit the observed behavior.

1.1. Model repair: between conformance checking and discovery

A more promising approach is to *repair* the original model such that it can replay (most of) the event log while staying close to the reference model (cf. Fig. 1(middle)). In [18], we introduced a new type of process mining: *model repair*. Like conformance checking we use a process model N and an event log L as input. If model N conforms to L (i.e., the observed behavior can be fully explained by the model), then there is no need to change N. However, if parts of N do not conform to L, these parts can repaired using the technique presented in this paper. Unlike discovery, the parts of the model that are not invalidated by the event log are kept as is. The resulting repaired model N' can be seen as a "synergy" of original process model N and event log L.

There are three main use cases for model repair:

- *Improving conformance checking diagnostics*: Conformance checking identifies discrepancies between modeled and observed behavior, e.g., by showing misalignments or places where tokens are missing during