Information Systems 56 (2016) 284–303

Contents lists available at ScienceDirect

Information Systems

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

BPMN Miner: Automated discovery of BPMN process models with hierarchical structure

Raffaele Conforti [a], Marlon Dumas [b], Luciano García-Bañuelos [b], Marcello La Rosa [a,c]

[a] Queensland University of Technology, Australia
[b] University of Tartu, Estonia
[c] NICTA Queensland Lab, Australia

ARTICLE INFO

Article history:
Received 15 December 2014
Received in revised form
7 May 2015
Accepted 9 July 2015
Available online 17 July 2015

Keywords:
Process mining
Automated process discovery
BPMN

ABSTRACT

Existing techniques for automated discovery of process models from event logs generally produce flat process models. Thus, they fail to exploit the notion of subprocess as well as error handling and repetition constructs provided by contemporary process modeling notations, such as the Business Process Model and Notation (BPMN). This paper presents a technique, namely BPMN Miner, for automated discovery of hierarchical BPMN models containing interrupting and non-interrupting boundary events and activity markers. The technique employs approximate functional and inclusion dependency discovery techniques in order to elicit a process–subprocess hierarchy from the event log. Given this hierarchy and the projected logs associated to each node in the hierarchy, parent process and subprocess models are discovered using existing techniques for flat process model discovery. Finally, the resulting models and logs are heuristically analyzed in order to identify boundary events and markers. By employing approximate dependency discovery techniques, BPMN Miner is able to detect and filter out noise in the event log arising for example from data entry errors, missing event records or infrequent behavior. Noise is detected during the construction of the subprocess hierarchy and filtered out via heuristics at the lowest possible level of granularity in the hierarchy. A validation with one synthetic and two real-life logs shows that process models derived by the proposed technique are more accurate and less complex than those derived with flat process discovery techniques. Meanwhile, a validation on a family of synthetically generated logs shows that the technique is resilient to varying levels of noise.

1. Introduction

Process mining is a family of techniques to extract knowledge of business processes from event logs [1]. It encompasses, among others, techniques for automated discovery of process models. A range of such techniques exist that strike various tradeoffs between accuracy and understandability of discovered models. However, the bulk of these techniques generate flat process models. When contextualized to the standard Business Process Model and Notation (BPMN), they produce flat

E-mail addresses: raffaele.conforti@qut.edu.au (R. Conforti), marlon.dumas@ut.ee (M. Dumas), luciano.garcia@ut.ee (L. García-Bañuelos), m.larosa@qut.edu.au (M. La Rosa).

http://dx.doi.org/10.1016/j.is.2015.07.004
0306-4379/© 2015 Elsevier Ltd. All rights reserved.

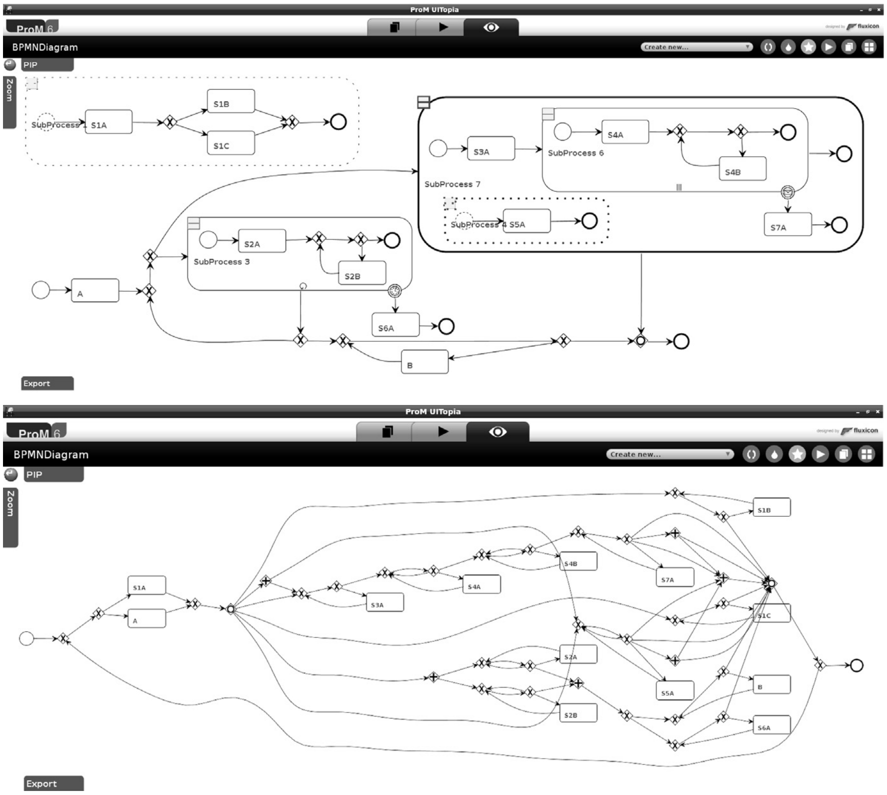

Fig. 1. BPMN model obtained with and without applying the proposed technique on a synthetic log of an order-to-cash process (using InductiveMiner to generate flat models).

BPMN models consisting purely of tasks and gateways. In doing so, they fail to exploit BPMN's constructs for hierarchical modeling, most notably subprocesses and associated markers and boundary events.

To fill the gap, this paper presents an automated process discovery technique – namely BPMN Miner – that generates BPMN models from event logs. In order to exploit BPMN's hierarchical modeling constructs, BPMN Miner incorporates mechanisms for detecting subprocesses, interrupting and non-interrupting boundary events, event subprocesses, and loop and multi-instance activity markers.

BPMN Miner takes as input an event log consisting of a set of event records, each including a timestamp, an event type (indicating the task that generated the event), and a set of attribute-value pairs. Such logs can be extracted from appropriately instrumented information systems [1]. For example, we validated the technique using logs with these characteristics from an insurance claims system and a grant management system, while Nooijen et al. [2] discuss a log with similar characteristics extracted from an Enterprise Resource Planning (ERP) system.

An example of a BPMN model discovered by BPMN Miner is shown at the top of Fig. 1. At the bottom is shown a flat BPMN model obtained from the Petri net discovered from the same log using the InductiveMiner plugin of the ProM framework [3].

The proposed technique exploits three key ideas to address the problem of identifying subprocesses and hierarchical relations: (i) that the set of events of an event type can be seen as a relational table; (ii) that event types sharing a common primary key are likely to belong to the same (sub-)process; and (iii) that foreign keys between event types are indicators of process–subprocess relations. Accordingly, BPMN Miner employs functional and inclusion dependency discovery techniques in order to cluster event types into groups corresponding to parent processes and subprocesses. Given the resulting process hierarchy, the technique splits the log into parent process logs and subprocess logs and applies existing process model discovery techniques to each log so as to produce a flat model for each node in the hierarchy. The resulting models and logs are analyzed heuristically to identify boundary events, event subprocesses and markers.

A major concern of process mining techniques in general, and automated process discovery techniques in particular, is dealing with *noise* [1]. In this setting, the term *noise* encompasses data quality issues – such as data entry errors, missing values or missing records – as well as infrequent behavior arising for example from one-off exceptions made during the execution of the business process. The presence of noise makes it difficult to derive readable process models, since each instance of noise may potentially generate additional paths in the discovered process model and the proliferation of such exceptional paths leads to spaghetti-like process models.