Information Systems 46 (2014) 123–139

Contents lists available at ScienceDirect

Information Systems

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

Bridging abstraction layers in process mining

Thomas Baier [a,*], Jan Mendling [b], Mathias Weske [a]

[a] Hasso Plattner Institute at the University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, D-14482 Potsdam, Germany
[b] Wirtschaftsuniversität Wien, Welthandelsplatz 1, 1020 Vienna, Austria

ARTICLE INFO

Available online 9 May 2014

Keywords:
Process mining
Abstraction
Event mapping

ABSTRACT

While the maturity of process mining algorithms increases and more process mining tools enter the market, process mining projects still face the problem of different levels of abstraction when comparing events with modeled business activities. Current approaches for event log abstraction try to abstract from the events in an automated way that does not capture the required domain knowledge to fit business activities. This can lead to misinterpretation of discovered process models. We developed an approach that aims to abstract an event log to the same abstraction level that is needed by the business. We use domain knowledge extracted from existing process documentation to semi-automatically match events and activities. Our abstraction approach is able to deal with n:m relations between events and activities and also supports concurrency. We evaluated our approach in two case studies with a German IT outsourcing company.

1. Introduction

Process mining finds increasing uptake in practice. Using the event data logged by IT systems, process mining algorithms discover and enhance process models or check whether the execution of a process conforms to specification [1]. Looking at conformance checking and enhancement of process models, it is obvious that the events stemming from IT systems have to be mapped to activities defined in process models. However, the events are typically more fine-granular than the activities defined by business users. This implies that different levels of abstraction need to be bridged in order to use these process mining techniques. Furthermore, such a mapping is necessary not only for conformance checking and process model enhancement, but also for discovery. The benefit of a discovered process model can only be fully exploited if the presented results are on an abstraction level that is easily understandable for the business user. Nevertheless, most current process mining techniques assume that there is a 1:1 mapping between events and activities. Only a few abstraction approaches address the mapping challenge by clustering events that can be bundled into singular activities (see e.g. [2–4]). However, these techniques have limited capabilities in dealing with complex mappings between events and activities and most often neglect n:m relationships and concurrency in the execution. Also, they provide no or only limited support for correctly refining these mappings based on the domain knowledge.

In this paper, we build on ideas presented in prior work [5,6] for tackling this mapping problem. Our contribution is a mapping approach that suggests relations between events and activities in an automated manner using existing process documentation as e.g. work instructions. For the set of suggested event–activity relations, we define means to dissolve n:m relations. In contrast to existing approaches, the method introduced in this paper is designed to deal with concurrency and to handle n:m relations between events and activities. We extend our

* Corresponding author. Tel.: +49 331 5509273.
E-mail addresses: thomas.baier@hpi.uni-potsdam.de (T. Baier), jan.mendling@wu.ac.at (J. Mendling), mathias.weske@hpi.uni-potsdam.de (M. Weske).

http://dx.doi.org/10.1016/j.is.2014.04.004
0306-4379/

previous work to deal with the complete life cycle of activities and to allow for zooming functionality in process discovery. Moreover, we introduce more sophisticated means to address the challenges of shared functionalities and loops. The capabilities of our approach are evaluated based on two case studies with a service outsourcing provider. The results demonstrate its benefits and emphasize the sensitivity of conformance and performance analysis to the defined mapping problem. Our approach can be used as a preprocessing step for every process mining technique, and therefore adds to the overarching field of business process analysis.

The paper is structured as follows. Section 2 describes the problem of different abstraction levels of event logs and process models. Furthermore, the preliminaries for our approach are introduced. Section 3 introduces the strategies to overcome the gap between abstraction levels of event log and process model. In Section 4, we show the results from case studies where we benchmarked our approach against manually created mappings, and outline the implications on conformance testing and performance analysis. Related work and prior research are reviewed in Section 5. Section 6 discusses the implications of our work on research and practice, and elaborates on current limitations. In Section 7 we conclude with a short summary of this work and give an outlook to future research.

2. Problem statement and preliminaries

This section provides an example to illustrate the research problem and introduces the preliminaries on which our approach builds.

2.1. Problem description

In order to make the problem more comprehensive, we will use the Incident Management process as defined in the IT Infrastructure Library (ITIL) [7]. Fig. 1 shows the process model at a very abstract level. At the bottom of Fig. 1, an excerpt of six cases from a corresponding event log is displayed. The different abstraction levels of process model and execution log spawn multiple challenges that need to be addressed in order to map the events to their corresponding activities.

The first challenge (1) is the diverging level of abstraction between events and activities and the effective usage of external knowledge to bridge the gap. While there are different approaches for abstracting event logs to a higher level, like [2–4], none of these approaches makes systematic use of external knowledge to map events to defined activities. Typically, organizations maintain detailed textual documentation of a process that extends the information provided in the model on a lower abstraction level. This knowledge should be leveraged to bridge the gap between the high-level process model and the low-level event log.

The second challenge (2) is the *unknown relation of events and activities*. In practical settings it is often not known a priori which events refer to which activity. An automated derivation of the relation between events and activities is non-trivial, as simple string matching techniques often do not work between different levels of abstraction. In the given Incident Management example for instance the two events “Group” and “Details” have to be related to the activity “Incident logging”. Existing abstraction approaches try to solve this challenge by clustering events that occur in temporal proximity [4]. Often this does not reflect the partitioning of activities as domain experts would expect it. Furthermore, different events also may have different meanings with respect to the life cycle of an activity. For instance, the event “Group” might always signal the start of activity “Incident logging”, while the event “Details” signals the end of the activity. This information is important when it comes to performance analysis and the calculation of activity durations. Last but not least, certain events might not be interesting for the aimed abstraction and should be filtered out in a convenient way.

The third challenge (3) is the *use of shared functionalities*, where different activities access the same functionality of the IT system. In this case, an event of the same type refers to multiple different activities. For example, Fig. 1 depicts the event “CI” that belongs to either one of the activities “Initial diagnosis”, “Investigation and diagnosis” or “Incident closure”. The abbreviation CI stands for the configuration item, i.e. the affected IT component. Depending on when the CI is changed during the process, the change of the CI refers to different activities. When happening at the beginning of the process

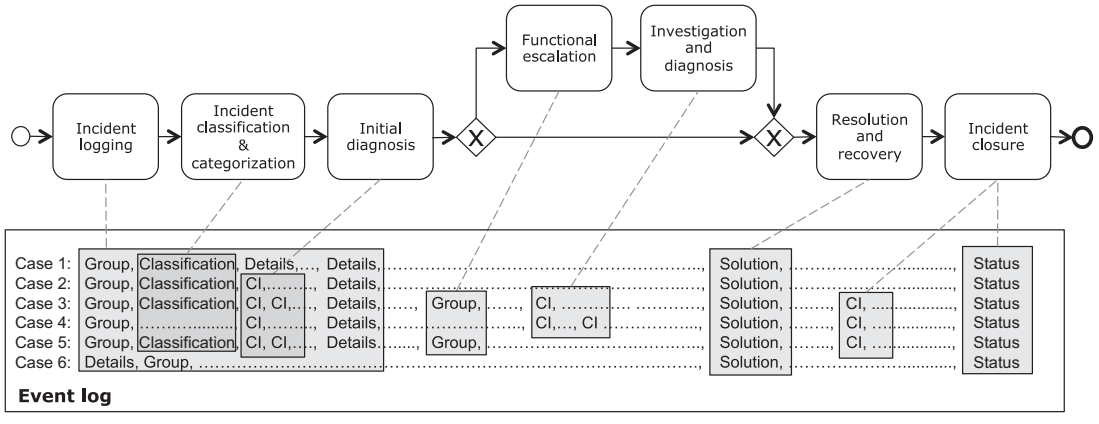

Fig. 1. Example of event to activity relations: Incident Management process model and low-level event log with shared functionalities and concurrency.