Information Systems 62 (2016) 42–59

Contents lists available at ScienceDirect

Information Systems

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

Do activity lifecycles affect the validity of a business rule in a business process?

Mario Luca Bernardi [a,b,c,d], Marta Cimitile [a,b,c,d], Chiara Di Francescomarino [a,b,c,d], Fabrizio Maria Maggi [a,b,c,d,*]

[a] University of Sannio, Benevento, Italy
[b] Unitelma Sapienza University, Roma, Italy
[c] Chiara Di Francescomarino, FBK-IRST, Trento, Italy
[d] Fabrizio Maria Maggi, University of Tartu, Tartu, Estonia

ARTICLE INFO

Article history:
Received 9 March 2016
Accepted 6 June 2016
Recommended by: D. Shasha
Available online 11 June 2016

Keywords:
Process discovery
Rule mining
Discriminative mining
Non-atomic activities
Activity lifecycle
Linear temporal logic

ABSTRACT

Traditional process mining techniques offer limited possibilities to analyze business processes working in low-predictable and dynamic environments. Recently, to close this gap, declarative process models have been introduced to represent process mining results since they allow for describing complex behaviors as a compact set of business rules. However, in this context, activities of a business process are still considered as atomic/instantaneous events. This is a strong limitation for these approaches because often, in realistic environments, process activities are not instantaneous but executed across a time interval and pass through a sequence of states of a lifecycle. This paper investigates how the existing techniques for the discovery of declarative process models can be adapted when the business process under analysis contains non-atomic activities. In particular, we base our proposed approach on the use of discriminative rule mining to determine how the characteristics of the activity lifecycles in a business process influence the validity of a business rule in that process. The approach has been implemented as a plug-in of the process mining tool ProM and validated on synthetic logs and on a real-life log recorded by an incident and problem management system called VINST in use at Volvo IT Belgium.

1. Introduction

Process mining techniques are nowadays applied in a large variety of domains and provide insights to manage, control and improve business processes [1]. The starting point for any process mining algorithm is an event log, i.e., a set of process executions (a.k.a. *process instances*) that are recorded when the process is still running and that can be used (on-the-fly or post-mortem) for business process analysis and monitoring.

Traditional process mining techniques, producing procedural process models, have two main drawbacks: (i) the produced models tend to be large and complex, especially in flexible environments where process executions involve multiple alternatives; (ii) they offer limited possibilities to guide the mining process towards specific properties of interest. For these reasons, recently, techniques for the discovery of declarative process models have been investigated. Here, the discovered process behavior can be described as a compact set of business rules even in complex environments where process executions involve multiple alternatives [2–17,57]. The advantage of mining declarative process models lies in the possibility of describing the mined process behavior in terms of generic

* Corresponding author.
E-mail addresses: mlbernar@unisannio.it (M.L. Bernardi), marta.cimitile@unitelma.it (M. Cimitile), dfmchiara@fbk.eu (C. Di Francescomarino), f.m.maggi@ut.ee (F.M. Maggi).

http://dx.doi.org/10.1016/j.is.2016.06.002
0306-4379/

rules that allow any path to be executed as long as these rules are not violated.

One of the main limitations of the existing process mining techniques for generating declarative specifications from logs is that they do not take into consideration that activities in an event log can be non-atomic and different events can be part of the lifecycle of the same activity. This is a crucial issue since, in many practical cases, activities cannot be considered as instantaneous but they are executed across a time interval and pass through a sequence of states of a lifecycle. An activity lifecycle is a sequence of transactional states (a.k.a. *event types*) assumed by an activity during its execution. An example of activity lifecycle is $\langle a_{assign}, a_{start}, a_{complete} \rangle$, where a is the performed activity and *assign*, *start* and *complete* are event types.

In this work, we extend the study conducted in [18] presenting an approach to discover declarative specifications from logs with a strong focus on activity lifecycles. In particular, using the information available in an event log, we try to identify business rules whose validity strongly depends on the characteristics of the activity lifecycles available in the log. The approach uses Declare as declarative process modeling language [19] and makes use of the notion of rule activation [20]. An activation of a business rule in a process instance is an event whose occurrence imposes, because of that rule, some obligations on the occurrence of other events in the same process instance. For example, for the business rule "every request is eventually acknowledged" each request is an activation, since the occurrence of a *request* forces an *acknowledgement* to occur eventually. In this case, the activation "request" becomes a fulfillment or a violation depending on whether it is followed by an "acknowledgement" or not. In our context, the notion of rule activation is crucial because we try to identify characteristics of the lifecycle of an activation that allow us to discriminate on whether that activation is a fulfillment or not. These characteristics are identified through discriminative rule mining techniques and are expressed, in turn, using Declare (Declare rules over the states of the lifecycle).

With respect to [18], in this paper, we provide several extensions. The first extension concerns how events belonging to the same lifecycle are grouped together in a process instance. The lifecycle identification can be performed through a FIFO-based [21] or an event correlation based [22] approach. The first one is a typical "conservative approach" first in-first out in which if a new upcoming event can be connected to two events occurred in the past and belonging to two different lifecycles, the priority is given to the one that occurred before. The second approach consists in connecting events to a lifecycle whenever they share common values for some data attributes. In this case, it is required that events in the log carry some data attributes. In our previous work, the lifecycle identification was implemented only using the first in-first out approach. Now, the prototype has been extended also to support the lifecycle identification through event correlations. As second extension, the discovery algorithm has been restructured as a multi processor algorithm to improve the prototype performance. The third extension concerns the implementation of a ProM[1] visualizer to present the mining results to the user through a graphical representation. In addition, a wider experimentation has been carried out, in particular, with the aim to show how correlations can better identify lifecycles with respect to the conservative approach when data is available and to demonstrate the performance improvement obtained using the distributed algorithm. Finally, a different approach for discriminative rule mining has been implemented. In our previous work, we provided the user with one of the possible discriminative rules having the same class probability (randomly chosen). Now, a sophisticated mechanism has been implemented to choose the discriminative rules (among the ones with the same class probability) that are the most interesting for the user (specifically, we choose the ones that are the strongest with respect to the logical implication).

The presented approach has been implemented as a ProM plug-in. This prototype has been used to validate our technique on both synthetic logs and a real life log recorded by an incident and problem management system called VINST in use at Volvo IT Belgium.

The rest of the paper is structured as follows. Section 2 provides a preliminary background about the Declare language, introduces the concepts of non-atomic activities and activity lifecycles and provides an overview on discriminative rule mining. Next, Section 3 illustrates the approach, based on the combination of declarative process mining algorithms (extended to consider non-atomic activities) and discriminative mining approaches. In Section 5, the experimentation is discussed. Section 6 discusses related work. Section 7 reports conclusions and future work.

2. Background

In this section, we introduce some preliminary knowledge needed to understand the techniques presented in this paper. In particular, in Section 2.1, we give an overview of the Declare language. In Section 2.2, we describe the transactional model for activity lifecycles defined in the XES standard. Finally, in Section 2.3, we give some background about discriminative rule mining.

2.1. Declare: declarative workflow modeling

Declare is a language for describing declarative process models first introduced in [19]. A Declare model consists of a set of rules applied to (atomic) activities. Rules, in turn, are based on templates. Templates are abstract parameterized patterns and rules are their concrete instantiations on real activities. Templates have a user-friendly graphical representation understandable to the user and their semantics can be formalized using different logics [23], the main one being LTL for finite traces, making them verifiable and executable. Each rule inherits the graphical representation and semantics from its template. The major benefit of using templates is that analysts do not have to

[1] www.processmining.org.