Decision Support Systems 100 (2017) 77–92

Contents lists available at ScienceDirect

Decision Support Systems

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

Discovering work prioritisation patterns from event logs

Suriadi Suriadi[a,*], Moe T. Wynn[a], Jingxin Xu[a], Wil M.P. van der Aalst[b,a], Arthur H.M. ter Hofstede[a,b]

[a]Queensland University of Technology, Australia
[b]Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Available online 8 February 2017

Keywords:
Resource behaviour mining
Queuing
Process mining

ABSTRACT

Business process improvement initiatives typically employ various process analysis techniques, including evidence-based analysis techniques such as process mining, to identify new ways to streamline current business processes. While plenty of process mining techniques have been proposed to extract insights about the way in which activities within processes are conducted, techniques to understand resource behaviour are limited. At the same time, an understanding of resources behaviour is critical to enable intelligent and effective resource management - an important factor which can significantly impact overall process performance. The presence of detailed records kept by today's organisations, including data about who, how, what, and when various activities were carried out by resources, open up the possibility for real behaviours of resources to be studied. This paper proposes an approach to analyse one aspect of resource behaviour: the manner in which a resource prioritises his/her work. The proposed approach has been formalised, implemented, and evaluated using a number of synthetic and real datasets.

1. Introduction

Business process management (BPM) enables organisations to improve the effectiveness and efficiency of their business operations by systematically documenting, managing, automating and optimising their business processes [1]. To achieve more with less, organisations need to focus on process efficiency, i.e., how their business operations could be improved. A plethora of literature and methodology exists on how one can improve process efficiency, e.g. Six Sigma [2]. However, as most business operations rely on human resources, e.g. employees, it is equally important to investigate whether these resources can be used in a more efficient manner; for example, how do employees spend their time between productive (e.g., waiting time) and non-productive (e.g., idle time) tasks? Are there any opportunities for increased resource utilisation?

Today's information systems record a wide variety of "events". Events may be generated by human behaviour (e.g., customers and employees), machines, and software. By leveraging state-of-the-art data analytics (including data mining, machine learning, and statistical techniques), valuable insights about resource behaviour can be extracted from this data to not only address the questions just presented, but also facilitate smarter resource management.

Within business processes, while resources are normally guided by business rules from the organisation and are constrained by the associated IT systems in terms of how they perform their work, resources typically have some freedom in prioritising their work, including the selection of activities (also known as *work items*) to perform and the order in which these activities are carried out. The way in which resources select the tasks to perform essentially forms the type of *queuing discipline* he/she applies. A *queuing discipline* refers to "the manner in which customers are selected for service when a queue is formed" [3]. The most common queuing discipline used in day-to-day life is the first-in-first-out style (FIFO) where work items that arrive first receive the highest priority, last-in-first-out (LIFO) where work items that arrive last receive the highest priority, and priority-based where priority is determined by some pre-determined rules.

The versatility of the concept of a queue has seen its application in many domains, from computer networks to business processes. Studies in the use of queues show that knowledge of queuing disciplines employed is important to design effective resource management strategy for ensuring appropriate staffing level [4,5] or performance stabilization [6,7]. Furthermore, studies show that queuing discipline may have a significant impact on the overall performance [8–13]. For example, the use of Shortest Process Time first discipline has

* Corresponding author.
E-mail addresses: s.suriadi@qut.edu.au (S. Suriadi), m.wynn@qut.edu.au (M.T. Wynn), j15.xu@qut.edu.au (J. Xu), w.m.p.v.d.aalst@tue.nl (W.M. vander Aalst), a.terhofstede@qut.edu.au (A.H. ter Hofstede).

http://dx.doi.org/10.1016/j.dss.2017.02.002

been shown to reduce cycle time as compared to FIFO in certain settings [11]. Within business processes, one can draw a parallel in how queuing discipline employed by resources can substantially impact overall process performance. For example, a predominantly LIFO work prioritisation behaviour of resources may very likely lead to a LIFO case completion trend - a phenomenon that is not desirable from a customer satisfaction perspective. The interplay between the assignment of work items to resources and their queuing discipline also impacts overall process performance. For example, assigning a work item involving *calling customers* to a resource who always prioritises the execution of *e-mailing customers* will easily lead to the building up of longer (and rather unfair) waiting times for the former task. This highlights an undesirable situation where the assignment mechanisms of work items to resources, and the choice of queuing discipline of the resources in the process are *out of sync*.

A clear understanding of resource behaviour can assist organisations in identifying undesirable (and perhaps unexpected) working patterns which will guide them in investigating contextual factors (e.g. the way in which a list of tasks is presented to users on their screens) that may inadvertently encourage the expression of such behaviours by employees, leading to a clear direction for process improvement (e.g. changing the default ordering of work items). As reported in this article, this is precisely one of the insights we extracted.

The scenarios above clearly demonstrate the importance of understanding resource behaviour: it allows one to identify individual resource behaviour (which may be problematic) and to understand their compound effects on overall process performance. Most importantly, insights about resource behaviour will nicely complement existing process improvement strategy, enabling intelligent adaptation of the way in which processes are designed (to achieve the best process outcomes) to the way in which resources tackle their tasks in the processes.

In this article, we present a new data-driven approach to learning the prioritisation order used by a resource to carry out the work items (in relation to a particular business process). As shown in Fig. 1 (left-hand-side figure), a business process is typically guided by a process model. A process model captures those activities that need to be performed, the temporal order in which they are to be executed (e.g. sequentially or in parallel), and the resource(s) who can execute the various activities in the process. The execution of various instances of a process is often recorded in transactional records (also known as *event logs*).

Event logs typically contain information about the activities (or work items) that have been executed, the time they occurred, and the identifiers of employees who carried out the activities. By combining process analysis and data mining techniques, the emerging discipline of *process mining* provides a collection of novel techniques to exploit and extract process-related insights from raw event data [14]. Research in the domain of process mining has traditionally been focused on *process discovery* (i.e., automated discovery of the control flow of a process from data attributes recorded in an event log), *conformance checking* (i.e., detection of where and how deviances in processes occurred by comparing observation seen in a log with normative process models or business rules), and *performance analysis* (i.e., identifying bottlenecks and extracting process performance metrics). Relatively few research studies have been conducted that focus on the resource perspective [15–20], and to our knowledge, none of these works focus on discovering resources work prioritisation order.

Our approach makes use of detailed transactional records of executed processes (i.e. event logs) to determine the queuing discipline employed by the resources (Fig. 1 - right-hand-side). Such a data-driven approach has the advantage of objectively exposing the actual way in which resources work, which may, and often do, contradict anecdotal wisdom or recommended business practices.

It is not our goal to monitor and control the way in which resources work. This paper is about discovering the work prioritisation patterns of resources and their effects on the overall process which can be performed in a privacy-preserving manner (see Section 5).

Our approach has been implemented as a plug-in for the open-source process mining tool ProM[1]. We evaluate the correctness of our approach and implementation by testing the tool using synthetic logs. We demonstrate the usefulness of our approach in a case study with an Australian-based insurance organisation. In particular, our case study manages to extract useful insights about behaviours of resources that may be useful for the stakeholders to design a more targeted actions.

The rest of the paper is organised as follows. Section 2 presents the proposed approach for learning work prioritisation patterns. Section 3 discusses a prototype implementation of the approach within the open-source process mining tool, ProM. Sections 4 and 5 present findings from the evaluation of the proposed approach using synthetic and real-life datasets. Section 6 summarises related work in the areas of organisational mining and queuing theory. Section 7 concludes the paper.

2. Learning work prioritisation patterns

A descriptive overview of our approach is provided in Section 2.1, and formalised in Section 2.2.

2.1. Approach

The proposed approach is illustrated in Fig. 2. The log shown at the top of Fig. 2 depicts a snippet of the events performed by two resources: Carol and Eliza. Each row in the log represents an event. For example, the first row of the log records an event capturing the assignment of a work item to the resource named Carol. The work item in this event is defined by the activity 'create PO' that needs to be executed for a particular process instance of which the identifier is '330'. As a short form, we give an identifier for the work item captured by every event in the log (e.g. C1 for the work item represented by the first event in the log).

By observing such an event log, we can build the worklist of a resource, ordered according to the times the work items are assigned to the resource (i.e., the in-list) and the corresponding (partial) list of work items completed by the resource, ordered according to the time the work items are completed (i.e., the out-list). For example, the bottom-left part of Fig. 2 depicts an in-list for resource Carol at a particular timestamp t'_3 (which happened *just immediately before* t_3) whereby three work items (C1, C2, and C3) have been assigned to her.

From this in-list, we build the *expected ordering of work items output* at time t_3 by assuming a certain queuing discipline. For example, if we hypothesise that Carol works on a FIFO basis, then we should expect the order in which the work items are completed to be the same as the order in which the work items were assigned.

The bottom-right side of Fig. 2 shows the out-list of Carol at time t_3, just after the completion of work item C3. Whenever we see a work item being completed, we first determine the expected work item that should be seen at the out-list based on the assumed queuing discipline and extract the in-list position of that work item. Next, we calculate the *distance* between the in-list *position* of the *expected* work item and the *in-list position* of the work item that *actually appears* in the out-list. For example, in Fig. 1, if Carol adopts the FIFO queuing discipline, the expected work item to be seen at time t_3 is C1 (which assumes the first position in Carol's in-list). However, if Carol adopts the LIFO discipline, the expected work item to appear in the out-list at time t_3 is C3 (which assumes the third position in

[1] www.promtools.org - Resource Queue Behaviour package.