Expert Systems with Applications 41 (2014) 7291–7306

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Merging event logs for process mining: A rule based merging method and rule suggestion algorithm

Jan Claes *, Geert Poels

Ghent University, Department of Business Informatics and Operations Management, Tweekerkenstraat 2, 9000 Gent, Belgium

ARTICLE INFO

Article history:
Available online 11 June 2014

Keywords:
Business process management
Process mining
Event log merging
Inter-organizational process modeling

ABSTRACT

In an inter-organizational setting the manual construction of process models is challenging because the different people involved have to put together their partial knowledge about the overall process. Process mining, an automated technique to discover and analyze process models, can facilitate the construction of inter-organizational process models. This paper presents a technique to merge the input data of the different partners of an inter-organizational process in order to serve as input for process mining algorithms. The technique consists of a method for configuring and executing the merge and an algorithm that searches for links between the data of the different partners and that suggests rules to the user on how to merge the data. Tool support is provided in the open source process mining framework ProM. The method and the algorithm are tested using two artificial and three real life datasets that confirm their effectiveness and efficiency.

1. Introduction

The awareness that organizations, in their attempts to optimize business processes, have to look beyond their organizational boundaries, exists in academia (Håkansson & Snehota, 1989; Legner & Wende, 2007) and in practice (Bernabucci, 2008; Grefen, Eshuis, Mehandjiev, Kouvas, & Weichhart, 2009). Numerous inter-organizational integration efforts are supported by the construction and analysis of business process models (Bolstorff & Rosenbaum, 2008; Ghattas & Soffer, 2009; Min & Zhou, 2002) and have proven their value (Höfferer, 2007; Van der Aalst, 1999a; Van der Aalst, 1999b; Van der Aalst, 2000).

When different organizations construct process models of their joint operations, this is called inter-organizational process modeling (Bouchbout & Alimazighi, 2011). In theory, there are many benefits to the alignment of processes across organizational boundaries (Bernabucci, 2008; Bolstorff & Rosenbaum, 2008; Cohen & Roussel, 2005; Håkansson & Snehota, 1989; Stadtler, 2008; Yu, Yan, & Cheng, 2001). Therefore, one would expect organizations to jointly organize their process at a large scale. However, in practice, it rarely happens that business partners share highly strategic data (Simatupang & Sridharan, 2001). Nevertheless, there are several cases where inter-organizational process modeling occurs in reality: (i) between organizations that are not competing (e.g., government, non-profit, healthcare), and (ii) for decentralized end-to-end processes within organizations (e.g., multinationals, shared service centers, multiple labels) (Hoogland, 2012).

The manual modeling of business processes is, however, time-consuming and prone to subjective decision-making, which provides good reasons to adopt automated modeling techniques such as process mining (Rozinat, Mans, Song, & Van der Aalst, 2009; Van der Aalst, 2008; Van der Aalst et al., 2003). Process mining makes use of recorded historical process data in the form of so called event logs to discover and analyze as-is process models (Van der Aalst, 2011). In an inter-organizational setting, these data are distributed over different sources, which each encompass partial information about the overall business processes. Currently available process mining techniques (e.g., Heuristics Miner (Weijters & Van der Aalst, 2006), Fuzzy Miner (Günther & Van der Aalst, 2007), Dotted Chart Analysis (Schonenberg, Weber, Van Dongen, & Van der Aalst, 2008), Conformance Checking (Rozinat & Van der Aalst, 2008), Social Network Mining (Van der Aalst & Song, 2004)[1]) require these data first to be merged into one structured dataset.

The recorded historical process data that are used as input for process mining techniques can be merged at three levels: raw data level (i.e., merging databases and/or files), structured data level (i.e., merging event logs), and model level (i.e., merging process

* Corresponding author. Tel.: +32 9 264 98 31; fax: +32 9 264 42 86.
E-mail addresses: jan.claes@ugent.be (J. Claes), geert.poels@ugent.be (G. Poels).

[1] These are the five most used process mining plug-ins in ProM 6 according to (Claes & Poels, 2013).

http://dx.doi.org/10.1016/j.eswa.2014.06.012
0957-4174/

models). When merging at structured data level, the individual partners of the inter-organizational process are responsible for selecting and structuring the data from their own information systems and by consequence in choosing the appropriate abstraction level and viewpoint of that part of the data. In case the data is merged at raw data level, the operation of structuring these raw data would form a bigger challenge, because then also the required knowledge about the meaning of and relations between the data should be brought together. On the other hand, the choice of a specific process mining technique can be postponed, because the result of merging the data at structured data level is an event log on which all existing process mining techniques (that use an event log as input) can still be applied. In the case of a model level merge, postponing the choice of mining technique to be employed would be impossible. Therefore, in the context of inter-organizational process modeling, the structured data level seems to be the appropriate level for merging the data.

There are, however, to date no techniques for merging the data at structured data level. In practice, process mining users nowadays turn to merging techniques at raw data level or at model level, but they report serious shortcomings in this way of working (Claes & Poels, 2013). Moreover, the shortage of suitable *tool support* for merging event logs (regardless the merge level) appears to be confirmed by a recent survey about the perception of the utility and usability of process mining techniques and tools (Claes & Poels, 2013).

Therefore, the research question that forms the basis for the research presented in this paper, is how to merge the data at structured data level of partners involved in an inter-organizational (business) process in such a way that the benefits of automated process mining are preserved (i.e., speed and objectivity). The significance of the research question is demonstrated by the recently published process mining community manifesto (Van der Aalst et al., 2011). In this manifesto researchers and practitioners in the field of process mining listed eleven important open challenges that need to be addressed to increase the applicability of process mining techniques. Challenge 1 is the merging (and finding and cleaning) of event data. Such merging is required for cross-organizational process mining, which is pointed out in challenge 7 as another important area of research. Two types of cross-organizational process mining are identified: (i) organizations collaborate to handle process instances together and (ii) different organizations execute the same process while sharing experiences, knowledge, or a common infrastructure (Van der Aalst et al., 2011). The contribution of this paper is a new method and algorithm with tool support to merge event logs (i.e., at structured data level) of different sources in support of inter-organizational process modeling.

The merging method that we designed to address the research question comprises two consecutive steps for the merge of two event logs:

(i) discover links between the two event logs to indicate which data in both event logs are considered to belong to the same process instance, and
(ii) based on the configured links, represented by merging rules, merge the data of both event logs to form a new event log.

The research described in this paper was performed using a design science approach (Hevner, March, Park, & Ram, 2004). Hevner et al. define design science research as "*research through the building and evaluation of artifacts designed to meet the identified business need*" (Hevner et al., 2004). The artifacts that are the subject of this paper are (i) an event log merging *method*, (ii) a merging rule suggestion *algorithm*, and (iii) their *implementation* in the process mining tool ProM. The algorithm searches for possible links between event logs and suggests merging rules to the user to support the merging method. Although in theory the method and its implementation (and the algorithm and its implementation) form separate artifacts, the large amount of involved data makes it practically impossible to treat them as separate artifacts that would have to be developed and evaluated separately. Hence, the evaluation of the method and algorithm is performed based on their implementations, i.e., the evaluation concerns the *implemented method* and the *implemented algorithm*.

The knowledge base on which our research is grounded is provided in Section 2. Methodical guidance for our research activities was found in the Design Science Research Methodology (DSRM) for Information System Research (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). This methodology requests the completion of six activities (see Fig. 1). The *Identification of the Problem & Motivation* is addressed by a recent process mining survey paper (Claes & Poels, 2013) that demonstrates the need for method and tool support for merging process mining data, which is also explicitly expressed by the process mining community in Challenge 1 and 7 of the process mining manifesto (Van der Aalst et al., 2011). The *Objectives of a Solution* were already defined in this introductory section. In short, the research aims at developing a way to merge data for process mining at structured data level. To be in line with the overall objectives of process mining, the merge of data in the form of event logs – as a preparatory step for process mining – needs to be accomplished quickly and objectively. Therefore, a high degree of automation is desired. The *Design and Development* of the artifacts is explained in Sections 3 and 4. In Section 3, the two steps of the *method* and its implementation in a well-known process mining tool are explained. Section 4 reports on the development of the rule suggestion *algorithm* and its implementation. The *Demonstration* and the *Evaluation* of the artifacts based on two artificial and three real life datasets is the subject of Section 5. The *Communication* of the research is obviously the objective of this paper, earlier iterations were published in previous work (Claes & Poels, 2011a; Claes & Poels, 2011b). Finally, Section 6 provides a discussion and conclusion.

2. Background

2.1. Process mining

The goal of process mining is "to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's systems" (Van der Aalst, 2011, p. 8). Three types of process mining techniques exist: (i) process discovery, (ii) process conformance, and (iii) process enhancement (Van der Aalst, 2011). Process discovery is concerned about how to construct a process model based on historical process data structured in event logs. Process conformance uses historical process data to check for deviations in the process with respect to a given process model. Process enhancement uses the historical process data to project more information on a provided process model (such as durations of and between activities or decision determinants).

An *event log* is a hierarchically structured file with data about historical process executions. Hence, this file contains data about several (possibly different) executions of the same process, which are used by process mining techniques. Mostly, this file has to be constructed by structuring raw process data that can be found in files or databases (e.g., SAP Audit Trail), into events and traces.

An *event* is the most atomic part of a specific process execution. Event data can typically be found in information systems under the form of status updates (e.g., from 'invoice sent' to 'invoice paid') or activity records (e.g., 'mail sent to customer'). Besides a name, events can have several other attributes to indicate for example