Towards Security-aware Process Mining

Rafael Accorsi and Thomas Stocker

Albert-Ludwigs-Universität Freiburg, Germany
{accorsi,stocker}@iig.uni-freiburg.de

Abstract. This paper reports on ongoing work towards a novel approach to process mining to support security audits in dynamic PAIS.

1 Introduction

Although workflows are largely employed in mission-critical process-aware information systems (PAIS) demanding strong security and privacy guarantees [6], appropriate security audit methods are missing [7]. Computer assisted audit techniques based upon process mining [10] cannot cope with the *security analysis* of *evolving* workflows [9]. This is due to: firstly, lack of fine-grained models that consider the different process configurations; secondly, the sole focus on the control flow, neglecting the data flows.

This paper reports on ongoing work on an extension of process mining to address the first of the aforementioned problems. Process mining generates a single model that consolidates all the different executions happening in the log file. This produces a coarse view of the underlying process. Trace clustering techniques act as a preprocessing step for process mining [4, 5, 8], thereby allowing for a fine-grained set of models. The idea is to group traces according to different characteristics and, subsequently, mine a particular set of clusters. While clustering allows for the selective reconstruction of traces, it still fails to mine the complete "history" (i.e. *evolution provenance*) of a business processes, identifying their diverse "tenancies" and how they differ.

We propose an approach for time-oriented log-clustering that is able to directly reflect the changing dynamics of a workflow's structure. Our approach clusters traces according to the timepoint where workflow instance has been triggered. In doing so, we obtain a chronological ordering of workflow tenancies which allows an auditor to appreciate the modifications that have been made on the original workflow, e.g. inserting or deleting activities leading to new instances.

Related work. Trace clustering algorithms are either *similarity-based* [5, 8] or *precision-based* [4]. Similarity-based approaches cluster traces according to a predefined threshold of trace similarity. Traces falling within

the threshold are considered "similar", otherwise they stand for a different cluster. The challenge here is to set a reasonable threshold for trace similarity that does not lead to over- and under-generalization. Precision-based approaches are iterative methods based on process mining. The idea is to select clusters according to the whether the mined workflow is a model for all the actual traces. Whenever a mined specification is not able to execute a trace (i.e. it is not a model for the trace), then a new cluster originates. This continues until each trace has a model.

In both approaches, clusters (and mined models) are loosely couple with each other and not expressive enough for security analysis. First, one cannot tell the extent to which a model relates to the other; second, the generated models are not organized chronologically. In security audits, it is necessary to have both features: establish the relationship between models records their provenance, allowing auditors to appreciate the modifications that led to a model; and since auditors usually sample data according to time [7], and cluster based on time to obtain a more precise view of the process instances in a frame of time.

2 Time-based Clustering Method

To detect workflow adjustments (i.e. changes), we determine the expected workflow behavior; anomalies are treated as indicators for adjustment operations. To identify such changes in the model, we analyze the evolution of distances between workflow activities. Model adjustments are understood as operations that change the way a process can be executed in the sense of adding or removing activities.

The distance between any two workflow activities within a trace is defined as the number of intermediate activities. In a fixed model, this distance remains constant over time if the activities are aligned and ranges in fixed boundaries conditioned by the minimum and maximum distance as shown in Fig. 1 in any other case. Workflow adjustments cause boundary variations for at least one activity pair. Monitoring distance progresses for all possible activity-pairs allows the identification of workflow tenancies.

Typical workflow behavior is determined on the basis of a parameter s (window size), that specifies the minimum number of traces used as "training" data and also defines the minimum cluster size. Sequentially processing a log file L consisting of traces $t_1, \ldots, t_n$, according to their timestamp, our algorithm uses a window of s traces for determining the typical workflow behavior in terms of boundaries $min_{i,j}$ and $max_{i,j}$ for the distance $d_{i,j}$ of any two successive activities a_i and a_j, in a trace.

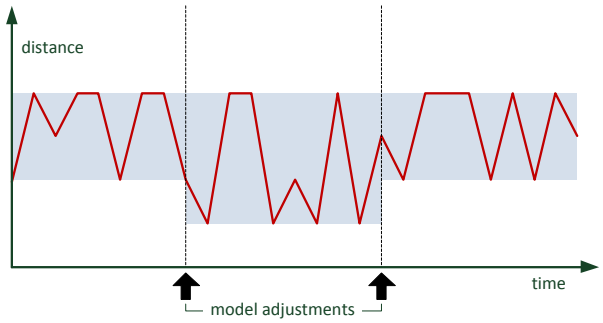

Fig. 1. Distance progress for a fixed activity pair.

Observed activity distances can continue their corresponding boundaries or introduce new ones. In detail, there are three cases to be distinguished:

1. The distance is equal to the boundary: $d_{i,j} = max_{i,j} \vee d_{i,j} = min_{i,j}$
2. The distance surpasses a boundary: $d_{i,j} > max_{i,j} \vee d_{i,j} < min_{i,j})$
3. The distance is within the boundaries: $(d_{i,j} < max_{i,j} \wedge d_{i,j} > min_{i,j}) \vee (d_{i,j} > min_{i,j} \wedge d_{i,j} < max_{i,j})$

Fig. 2 shows the intervals that are introduced in these three cases. In Case 1 nothing has to be done. While Case 2 definitely introduces a new interval, as old boundaries cease to hold, in Case 3 further analysis is necessary. If $d_{i,j}$ belongs to a new, smaller interval, the next s distance values reflect these new boundaries. Our algorithm uses a s-size lookahead to check this property. As long as the observed pair-distances of following traces do not conflict with the typical boundaries, they are put in the same cluster. Once a new interval is detected, the typical behavior is calculated again on basis of the next s distance values.

In going so, we identify modifications and their timepoint, thereby allowing time-based clustering. Instead of extracting a single workflow model from log data or clustering by similarity, time-based clustering enables auditors to decide about which models fall into a testing period and can assist in identifying relevant starting points for analysis.

3 Summary

Overall, the approach we put forward contributes to the following areas: *Business provenance* increases the traceability of end-to-end business operations in a flexible and cost-effective manner [3]. Our clustering technique and the subsequent mining provide provenance information.

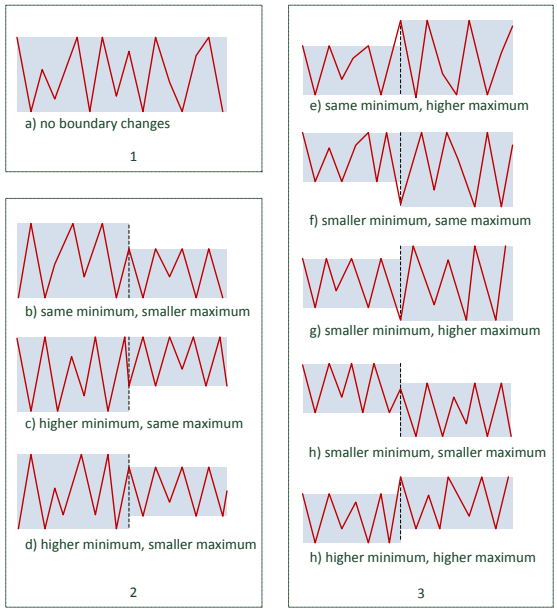

Fig. 2. Possible boundary changes.

Audit reduction provides filtering techniques that allow an auditor to select which of the various mined workflow specifications are to be audited.

Security audits use InDico to test mined workflow models for security properties [1]. InDico and the clustering methods we propose are realized in SWAT [2], a security workflow analysis toolkit. Further work extends SWAT to support the subsequent process mining.

References

1. R. Accorsi and C. Wonnemann. Strong non-leak guarantees for workflow models. In *ACM SAC*, pages 308–314. ACM, 2011.
2. R. Accorsi, C. Wonnemann, and S. Dochow. SWAT: A security workflow toolkit reliably process-aware information systems. In *Security Aspects of Process-aware Information Systems*. IEEE, 2011 (to appear).
3. F. Curbera, Y. N. Doganata, A. Martens, N. Mukhi, and A. Slominski. Business provenance - A technology to increase traceability of end-to-end operations. In *OTM*, vol. 5331 of *LNCS*, pages 100–119. Springer, 2008.
4. A. K. A. de Medeiros, A. Guzzo, G. Greco, W. van der Aalst, A. Weijters, B. F. van Dongen, and D. Saccà. Process mining based on clustering: A quest for precision. In *BPM Workshops*, vol. 4928 of *LNCS*, pages 17–29. Springer, 2008.

5. G. Greco, A. Guzzo, L. Pontieri, and D. Saccà. Discovering expressive process models by clustering log traces. *IEEE TKDE*, 18(8):1010–1027, 2006.
6. L. Lowis and R. Accorsi. Finding vulnerabilities in SOA-based business processes. *IEEE Transactions on Service Computing*, 2011 (to appear).
7. A. Sayana. Using CAATs to support is audit. *Inf. Systems Control J.*, 1, 2003.
8. M. Song, C. Günther, and W. van der Aalst. Trace clustering in process mining. In *BPM Workshops*, vol. 17 of *LNBIP*, pages 109–120. Springer, 2008.
9. R. Teeter and M. A. an Miklos Vasarhelyi. Remote auditing: A research framework. *Journal of Emerging Technology in Accounting*, to appear.
10. W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and T. Weijters. Workflow mining: A survey of issues and approaches. *Data Knowledge Engineering*, 47(2):237–267, 2003.