

Opportunities for Process Improvement: A Cross-Clientele Analysis of Event Data Using Process Mining

R.P. Jagadeesh Chandra Bose^(✉), Avantika Gupta, Deepthi Chander,
Ajith Ramanath, and Koustuv Dasgupta

Xerox Research Center India, Bangalore 560103, India
{jagadeesh.prabhakara,avantika.gupta,deepthi.chander,
ajith.ramanath,koustuv.dasgupta}@xerox.com

Abstract. Services organizations are always under pressure to operate under tight costs and to improve their operational efficiency. Transaction processing is one of the major operations in a services organization. An organization is typically trained to serve a standard set of processes within different domains across several clients. Although each client has their own specifics with respect to a process, there is a lot of commonality within similar processes across clients. An organization's operational KPIs (i.e., Key Performance Indicators like processing time) across these clients when dealing with such related processes might not be similar; an organization might perform well for some clients and perform below par on others. There is a need to gain insights for such variance in performance and seek opportunities to learn from well performing client engagements (e.g., establish best practices) and leverage these learnings/insights on non-performing clients. We present a framework for analysing operational event data of related processes across different clients to gain insights on process executions. We present results of analyzing real-world transaction processing operations of a large services organization using the proposed framework. Our analysis shows that resource workload, clarity of process definitions, experience, and skill proficiency are key factors that influence the average processing time of transactions.

1 Introduction

Services organizations cater to a large number of clients on a daily basis. Each client specifies certain Service Level Agreements (SLAs) to quantify performance, e.g., turnaround time, cost, quality etc., which need to be met by service providers. Service providers typically implement/deploy a service delivery framework to meet these SLAs. Most service providers are challenged by a constant need to closely monitor the performance and efficiency of their operations to meet stringent compliance requirements, handle cost pressures, inefficient processes and complex workflows. Inability to meet SLAs due to inefficient business processes can amount to 20% loss for businesses today.

Many of today's service organizations record event data pertaining to their operations, which have been exploited to gain insights on process executions, for example, using process mining [1] techniques. Traditionally, such insights are obtained in silos, i.e., processes pertaining to clients are analyzed independently. In recent years, there has been considerable interest in cross-clientele analysis. The main motivation for such an analysis stems from the fact that - although service providers may be performing variants of a similar process across its clients, the Key Performance Indicators (KPIs)¹ may vary significantly across clients. For instance, two clients requiring a similar process (e.g., document verification) to be executed, may incur very different turnaround times for process completion. In some cases, this can even result in the service provider meeting SLA specifications for some clients, and violating those of others, despite the similarity in processes executed for the clients. This variation in business process performance can be attributed to variations in workflow design, resource allocation, context dependencies, skill deficiencies, etc. Given that services organizations frequently encounter the need to execute similar processes across its clients, a siloed approach of operational execution data analysis does not suffice. There is a need to gain insights for such variance in performance and seek opportunities to learn from well performing client engagements (e.g., establish best practices) and leverage these learnings/insights on non-performing clients.

In this paper, we propose an extensible framework using process mining that supports cross-clientele learnings based on analysis of event logs containing business events of various clients and their processes. The framework comprises of

- an analytic engine that provides insights on various aspects of a process such as process discovery, conformance checking (e.g., deviations from expected behaviour), and performance analysis (e.g., turnaround times, working times, bottlenecks)
- a root cause diagnostic engine that assists in identifying the potential causes of some observed behaviour
- a recommendation engine that provides prescriptions for improving the operations of an organization.

We present some insights from analyzing the operational data of transaction processing pertaining to a large services organization. In particular, we demonstrate how the proposed framework can elicit factors that influence the performance of key client processes.

The remainder of the paper is organized as follows. Related work is discussed in Sect. 2. Section 3 provides some background on process mining and event logs. The framework for cross-clientele analysis is presented in Sect. 4. Section 5 presents and discusses the application of the proposed framework on a real-life case study. Finally, Sect. 6 concludes the paper.

¹ Note that SLAs typically are external metrics (for clients) while KPIs are internal metrics (for the organization).

2 Related Work

While most prior works in process mining adopt a client-specific view, recent works have recognized the benefits of process analysis across clients [5, 6]. Buijs et al. [5] cross-correlate process models of different clients with respect to their actual behaviours observed from event logs along process model quality metrics (e.g., process complexity), behavioral quality metrics (e.g., throughput time) and comparison metrics (e.g., process similarity). However, they do not provide any root cause analysis of the observed behavior. Process-related risks pertaining to deadline transgressions or overruns are inferred in [12] using statistical techniques by identifying potential risk indicators such as abnormal activity execution/waiting time, multiple executions of activity etc. Root cause analysis of deviant process instances are studied in [8, 11, 15]. The basic idea in all of these is to define some features (e.g., workload, sequence of activities, etc.) and translate the problem of finding root-causes into a classification problem.

The impact of factors such as the complexity of work, priority or importance of work and expertise of the worker, on the operational productivity of the worker have also been studied [14]. However, [8, 11, 12, 14, 15] all focus on analyzing a single client/process. In contrast, our work analyzes several clients/processes and in addition, we explore the influence of factors such as resource skills/capabilities, their proficiency, team composition and process complexity on various behavioral aspects of process executions. Our work presents a holistic, data-driven framework with pluggable components which analyzes, infers and identifies opportunities for process improvements based on cross-learnings from event log data pertaining to various client process executions.

3 Background

In this section, we provide some background on process mining. Process mining serves as a bridge between data mining and business process modeling and analysis. The starting point of process mining is the notion of an event log. An event log captures the manifestation of events pertaining to the instances of a single process. A *process instance* is also referred to as a *case*. Each event e in the log corresponds to a single case and can be related to an *activity* or a *task* (for instance, e corresponds to audit task). Events within a case need to be *ordered*. An event may also carry optional additional information like *timestamp* (e occurred on April 2nd 2015 at 13:25:10 IST), *resource* (e was executed by Kathy), *transaction type* (e is a start event, i.e., the start of the auditing task by Kathy), and various data elements such as *costs* (e costed \$0.5), etc. Timing information of when an event occurred is required to analyze the performance related aspects (e.g., waiting time, processing time, etc.) of the process. Resource information such as the person executing the activity is useful when analyzing the organizational perspective, and in conjunction with time, can help in analyzing the productivity of various resources. We refer to these additional properties as *attributes*. To summarize,

- an event log captures the execution of a process.
- an event log contains process instances or cases.
- each case consists of an ordered list of events.
- each event can be associated exactly to a single case.
- events can have attributes such as activity, time, resource, etc.

Event logs from different systems and organizations can be stored in different formats, e.g., databases, plain text files, etc. For process mining applications, a common event log format based on a process meta model, called the MXML format (**M**ining **X**ML) [7], has been proposed. This has been followed by a more recent advancement, called the XES (**eX**tensible **E**vent **S**tream) [9]. XES is adopted as the standard by the IEEE Task Force on Process Mining.

A process should be analyzed holistically across four dimensions: (i) control-flow, (ii) data, (iii) resource, and (iv) time. Note that these dimensions are not orthogonal. The topics in process mining can be broadly classified into three categories (i) *discovery*, (ii) *conformance*, and (iii) *enhancement*.

- *Discovery*: Process discovery deals with the discovery of models from event logs. These models may describe control-flow, organizational aspects, time aspects, etc. There are dozens of process discovery techniques generating process models using different notations (Petri nets, EPCs, BPMN, heuristic nets, etc.) [1].
- *Conformance*: Conformance or compliance checking deals with comparing an apriori model with the observed behavior as recorded in the log and aims at detecting inconsistencies/deviations between a process model and its corresponding execution log. In other words, it checks for any violation between *what was expected to happen* and *what actually happened*. The apriori models can be specified using different notions such as Petri nets, Declarative models, or in the form of business rules.
- *Enhancement*: Enhancement deals with extending or improving an existing model based on information about the process execution in an event log. For example, annotating a process model with performance data to show bottlenecks, throughput times, etc., by exploiting the timestamps in the event log.

For a classic introduction to process mining, the reader is referred to [1]. We adopt several techniques from process mining and the process mining tool ProM [16]² to discover insights on process executions. While ProM supports analysis of one event log pertaining to a process, we built a plug-in in ProM that analyzes a collection of logs together. In addition, we built a root cause diagnostic engine to analyze the causes for observed behavior.

4 Framework for Cross-Clientele Analysis

In this section, we discuss an extensible framework for performing cross-clientele analysis of operations within a services organization. Figure 1 depicts the framework for cross-clientele analysis. The basic building blocks of the framework are discussed below:

² See www.processmining.org for more information and to download ProM.

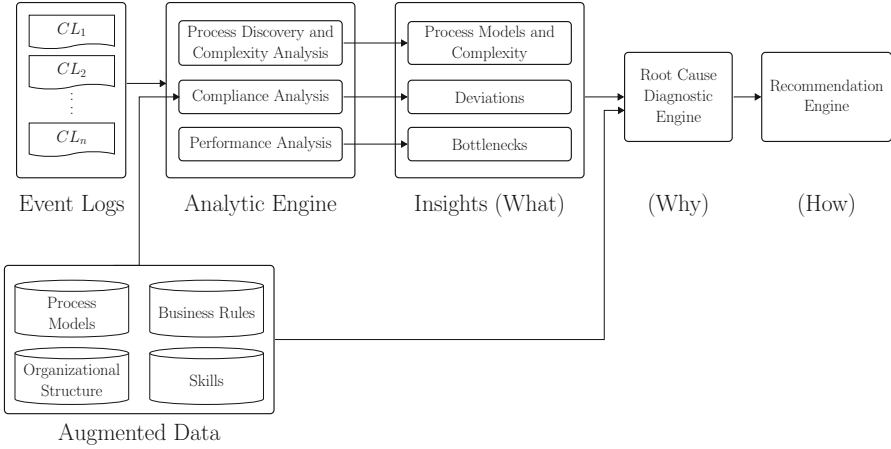


Fig. 1. Framework for analysis of process executions across clients.

- **Event Logs:** At the outset, we need to consider event logs pertaining to the operations of a particular process for all clients we are interested in analyzing. The particular choice of clients and processes can be driven from various means, e.g., one can choose clients across different geographies, based on their type such as gold, silver, and platinum, based on their KPI metrics (clients whose KPIs are met vs. those that are not), or can even be random.
- **Augmented Data:** In addition to the event logs, one can have other forms of data that augment the event logs. For example, one can have a repository of client process models corresponding to how the organization expects the process executions to happen, organizational structure eliciting the resources, their roles, departments, groups etc., data about the skills and proficiency of resources, business rules, etc. Some example business rules include: activity X should be executed by a resource with a proficiency at least 2 in skill S, activity Y should be executed within 2 h of executing activity X.
- **Analytic Engine:** Given a set of event logs and optional augmented data as mentioned above, one can derive several kinds of insights related to the process executions. The analysis techniques can be broadly divided into three categories:
 - *Process Discovery and Complexity Analysis:* This corresponds to the discovery of process models from event logs and analyzing them. We can compare how the process models vary across the clients, identify what is common between them, which workflow patterns are used in the processes, which client processes are complex (e.g., structural complexity), etc.
 - *Compliance Analysis:* This corresponds to verifying how compliant are client process executions with respect to their expected behavior. This takes as input event logs (capturing process executions) and process models or business rules (capturing expected behavior). One can apply replay techniques (replaying process instance traces onto the process models/

business rules) to identify what sort of deviations manifest, how often do they manifest in the process executions. If all client processes follow a similar process model, this gives a snapshot view of how compliant various client executions are w.r.t the expected behavior.

- **Performance Analysis:** This corresponds to various performance related aspects pertaining to the process executions, e.g., the average working/processing time of activities, the waiting times, sojourn times between activities, the turnaround time of the process etc. In addition, we perform resource performance analysis in the process, i.e., how do different resources execute an activity: the working/processing times that different resources take to execute an activity, deviations in execution times, etc.

Note that as mentioned above, insights should ideally cater to all the four perspectives (control-flow, data, resource, and time) of a process.

- **Root Cause Diagnostic Engine:** The insights that are uncovered in the earlier steps provide descriptive information as to what is happening in the operations of the process. Having discovered any interesting insights, the organization would be interested in identifying what the underlying (root) causes are for the observed behavior. The root cause diagnoser assists in identifying and corroborating factors that can be attributed to an observed behavior. The factors that influence the behavior of a process, e.g., resource skills and proficiency, process complexity, etc., can be (pre)defined. From the event logs and other augmented data sources, we extract the values corresponding to these features and the problem of finding the root causes can be posed as a learning problem. For example, for a given process, if we are interested in dissecting whether resource skill levels attribute to the variations in performance (working time), we can create a resource-skills matrix with each cell (i,j) in the matrix capturing the proficiency level of resource ‘i’ in skill ‘j’. We can classify the resources into various classes, e.g., efficient and poor, based on their performance. We can then apply techniques like decision trees to learn the discriminatory features if any between the defined classes. Simple correlation analysis techniques can also reveal any potential causes.
- **Recommendation Engine:** This corresponds to providing prescriptions to the organization towards improving their operations. The recommendations are driven by the root causes discovered in the earlier step, e.g., if skill proficiency is identified as a cause, one can recommend training of resources to improve their skill levels.

Note that the proposed framework can be extended with additional techniques. In this paper, we consider only the analytic and root cause diagnostic engines. The recommendation engine is beyond the scope of this paper. We are in discussions with the business stakeholders to translate/leverage some of our root causes to recommendations/prescriptions.

5 Cross-Clientele Analysis for Transaction Based Outsourcing (TBO) Business

In this section, we discuss the results of applying the proposed framework on the event logs pertaining to a transaction processing business organization within Xerox Services. The proposed framework has been implemented as a plug-in in the process mining tool ProM [16].

5.1 Business Context

For confidentiality, we anonymize the discussion on the business without losing any information. The organization serves clients across different domains $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$. Within each domain, D , there are several processes $\mathcal{P}^D = \{P_1, P_2, \dots, P_m\}$. For each process, P , the organization has defined a set of steps that caters to the process. For a particular client C and a process in a domain, P^C , the organization configures the steps to cater to the needs of the client. These configurations result in different transaction types $\mathcal{T}_C^P = \{T_1, T_2, \dots, T_k\}$ for the client process.

The resources in the organization are skilled on the defined steps of a process. Different resources are skilled to execute different processes and the proficiency levels of resources on the skills can vary.

The organization has around 1400 employees and provides services across 4 primary domains and 163 processes for over 90 clients. The organization typically handles between 5000 and 10000 transactions per day. Although the daily volume of transactions is relatively moderate, we look at three months of longitudinal data to demonstrate the efficacy of our framework. We believe that the proposed framework is capable of handling scenarios with much larger volumes of transactions.



Fig. 2. An expected process behavior for the transactions.

The organization receives instances of a process as transactions. Each transaction refers to a process instance and the typical workflow of a process corresponds to the transaction being assigned by a “team lead” to a resource whose role is termed “processor”. The processor then works on the transaction (based on a well defined set of steps). The processor can either finish the transaction in one go or can process it for some time, suspend it and later resume it for completion. Once the processor completes his/her task, the transaction is then assigned to an “auditor” for quality check. The auditor checks if the processor has processed the transaction correctly and if no issues are perceived, the transaction is completed. If issues are found, the auditor sends the transaction back to the processor, who has to rectify the errors identified. Just like the processor, an auditor can perform his/her task in one go or can do the task for some

time, suspend it and later resume it for completion. The organization expects the transactions failing the auditor checks to be a rare event. Figure 2 depicts the process behavior for transaction processing.

5.2 Data Set

We have selected four processes that are commonly executed by five clients in domain A .³ These are some of the most critical processes that the organization provides services on and the chosen clients are among the big ones. The organization is interested in gaining insights on how these processes are managed across these clients. Let these processes be denoted by P_1^A , P_2^A , P_3^A , and P_4^A and the clients by C_1^A , C_2^A , C_3^A , C_4^A , and C_5^A . For these client-process combinations we have considered all transactions that have started and completed over a three month period between January 2015 and March 2015.

Table 1 depicts the number of transaction types and the average number of steps per transaction type for each of the client-process combinations. Also depicted in Table 1 is the minimum and maximum number of steps among the transaction types for a client-process combination. We can see that clients C_1^A and C_2^A have a lot of heterogeneity in their processes (this is reflected in the large number of transaction types for processes P_1^A , P_3^A , and P_4^A). At the same time, several of these transaction types are simple (involving only few steps), which is reflected in the relatively low average number of steps for these processes when compared to the other clients. The organization is particularly interested in the way how processors are executing their task. Hence in the rest of the discussion, we focus on insights related to processors. Table 2 depicts the volume of transactions, the number of “processors” who worked on these transactions, and the average number of transactions per resource for each of the client-process combinations. Also highlighted in the table are some aberrations. For example, we can see that client C_4^A exhibits drastically different characteristics when compared to others, the average number of transactions per resource is much larger (278.82) when compared to other clients for P_1^A and much lesser (4.22) for P_3^A . Similarly, for client C_5^A , the average number of transactions per resource is less (6.75) for P_3^A and much larger (429.57) for P_4^A .

For each of these transactions, the organization records event data in the form of a worklog. The worklog contains information about the high level steps of the process elicited above, e.g., when a transaction arrived, the team lead who assigned the transaction to a processor, when it was assigned, and to whom it was assigned etc. We process this data and generate XES/MXML event logs amenable for process mining. The next few sections provide a detailed analysis using the proposed framework in Fig. 1

5.3 Process Discovery and Complexity Analysis

As illustrated in the framework in Fig. 1, several types of insights can be obtained using various analysis techniques. Figure 3 depicts the process model discovered

³ Recall that the organization serves clients across different domains.

Table 1. The number of transaction types (#TT), the average number of steps per transaction type, the minimum, and the maximum number of steps among the transaction types for different processes across different clients in domain A.

	P_1^A				P_2^A				P_3^A				P_4^A			
	#TT	avg.	min.	max.	#TT	avg.	min.	max.	#TT	avg.	min.	max.	#TT	avg.	min.	max.
C_1^A	88	1.5	1	11	2	4.5	1	8	11	1.45	1	6	34	2.5	1	9
C_2^A	53	5.0	1	29	5	32.4	4	53	22	1.9	1	8	9	3.0	1	6
C_3^A	13	5.2	1	15	4	5.75	1	9	5	6.2	2	11	5	6.0	4	8
C_4^A	7	5.0	1	11	1	1.0	1	1	3	1.3	1	2	-	-	-	-
C_5^A	9	8.2	6	12	2	8.5	7	10	7	2.4	1	6	2	6.5	6	7

Table 2. The number of transactions (#T), the number of processors (#R) deployed, and the average number of transactions per processor for different processes across different clients in domain A.

	P_1^A			P_2^A			P_3^A			P_4^A		
	#T	#R	avg.	#T	#R	avg.	#T	#R	avg.	#T	#R	avg.
C_1^A	2917	19	153.52	1323	9	147	599	22	27.22	1804	15	120.66
C_2^A	7146	36	198.5	600	23	26.09	404	20	20.2	3052	38	80.31
C_3^A	1736	12	144.66	440	16	27.5	965	21	45.95	1572	20	78.6
C_4^A	6413	23	278.82	38	9	4.22	201	7	28.71	-	-	-
C_5^A	1494	8	186.75	54	8	6.75	444	14	31.75	6014	14	429.57

using the heuristics miner [17] for client C_3^A and process P_1^A . We can see that the workflow followed is along the expected process behavior illustrated in Fig. 2. In addition to the control-flow, the discovered process model provides insights about how often a particular activity and flow was executed. We can see that for roughly 3.8% of the transactions, processors have suspended and resumed the processing of the transaction. Similarly for 1.8% of the transactions, an auditor has suspended and resumed the auditing of the transaction.

Figure 4 depicts the process model discovered using the heuristics miner for client C_4^A and process P_1^A . In this model, we can see that for 72 transactions, an auditor has found a quality issue with how the transaction is processed by a processor. For a significant number of transactions 16% (unlike the earlier case where it was just 3.8%), the processor has suspended and resumed the transaction. Furthermore, the processors have (re-)suspended the transaction several times (263 instances) before they finally complete it. This clearly indicates that the processors had difficulty in executing this transaction (as is evident both from the number of times quality issues have been discovered and the number of times processors have to suspend and resume the transactions). *One reason for this could be attributed to the relatively large volume of transactions that a resource is assigned on average.* Recall from Table 2 that for this client-process, the average number of transactions per resource (278.82) is much larger than

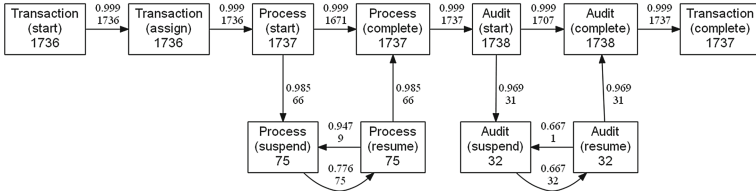


Fig. 3. Process model discovered using heuristics miner for client C_3^A and process P_1^A .

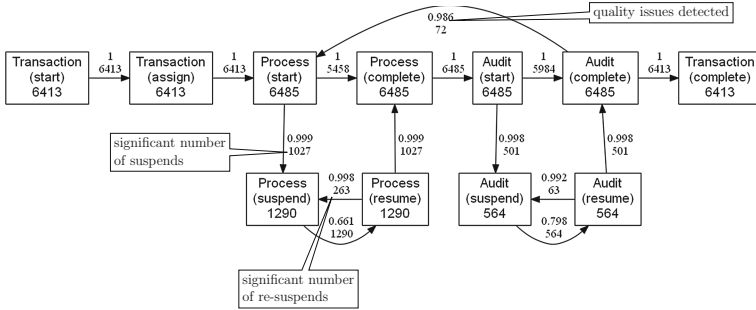


Fig. 4. Process model discovered using heuristics miner for client C_4^A and process P_1^A .

the average for this process across other clients. Process discovery techniques of our framework have thus been able to uncover the operational process model, even while providing insights w.r.t. the variants in control-flow aspects across client-process combinations.

5.4 Compliance Analysis

We next discuss some results on conformance checking (or compliance) analysis of the event data. Recall that conformance checking analyzes how conformant are real executions with respect to some expected behavior. One can specify the expected behavior in the form of a process model and replay techniques such as [2, 4, 13] can be applied. These techniques provide an objective *fitness* metric indicating how conformant a process instance is with respect to the expected behavior. In addition, they also reveal what deviations from the expected behavior are manifested in the process instance. One can also specify the expected behavior in the form of rules, for example as linear temporal logic (LTL) constraints for the control-flow perspective of a process and techniques such as [3, 10] can be used to identify the conformance or non-conformance of process instances. We have used the Petri net equivalent of the process model depicted in Fig. 2 as the model for expected behavior and replayed the event logs of each client-process using [4]. Table 3 depicts the results of conformance analysis where we elicit various classes of deviations uncovered, the client-processes, and the number of transactions where those deviations are manifested.

Table 3. Deviations uncovered during conformance checking analysis for the various client-processes of domain A.

Deviation	Client, Process	#Trans.
Transaction is assigned to two different processors	(C_2^A, P_2^A)	1
	(C_4^A, P_3^A)	1
Auditing is done by two different resources	(C_1^A, P_1^A)	1
Auditing is done after a transaction is completed	(C_3^A, P_1^A)	1
	(C_3^A, P_2^A)	5
	(C_3^A, P_4^A)	1
Quality issues are identified and processing is done more than once	(C_3^A, P_1^A)	1
	(C_4^A, P_1^A)	72
Transaction is withdrawn after processor completes the task	(C_3^A, P_2^A)	1

There are two instances where a transaction is assigned to two different processors. In one instance, (C_2^A, P_2^A) , the team lead first assigned the transaction to a processor who does not respond by accepting the transaction. The team lead then reassigns the transaction to a different processor. The other instance, (C_4^A, P_3^A) , presents an interesting scenario where the transaction is assigned to a different processor after the initially assigned processor has started working on the transaction. Deviations are observed in the audit executions as well. In one instance, auditing is done by two different auditors. The first auditor started auditing and suspended the task. It was later resumed by another auditor who completed the task. There are several instances where auditing was done after the transaction was completed. It is interesting to note that this deviation emanates only from operations related to client C_3^A across three different processes.

Recall from our earlier discussion that the organization expects quality check failures to be a rare event. We encountered two client-process combinations, (C_3^A, P_1^A) and (C_4^A, P_1^A) , where quality issues are perceived. Only one transaction failed the quality check in the former while 72 transactions failed the check in the latter (also as illustrated in Fig. 4). It is interesting to note that the quality issues are perceived only in process P_1^A . In one instance, (C_3^A, P_2^A) , a transaction is withdrawn after the processor finished executing the task. The deviations thus uncovered could then be analyzed by team leads and steps could be taken to prevent such deviations in the future.

5.5 Performance Analysis

We next discuss the results of performance analysis. Table 4 depicts the average and standard deviation of the working time that a processor takes to complete his/her task for the various client-process combinations. *The large deviations can be attributed to the variations in the number of check items (steps) among*

the transaction types of a client-process and also to the resources working on the process. We can see that processors take relatively very long time to finish tasks for all processes pertaining to client C_4^A as highlighted in the table. Note that P_2^A is defined over only one step for C_4^A and P_3^A over one/two steps unlike other clients. This implies that the entire process is defined as one/two steps rather than splitting into several smaller steps. The processor working on the transaction has to remember all that has to be done for C_4^A for this transaction using only these one/two check items (steps), which is difficult. Performance analysis can help us uncover bottleneck processes/tasks.

5.6 Root Cause Analysis

We analyzed if the number of steps in which processes are defined can *potentially be a root cause* for the variation in performance. The premise is that the more the number of steps, the finer the detail to which the process is explained.

Table 4. The average and standard deviation of working times (in seconds) taken by processors for different processes across different clients in domain A.

	P_1^A	P_2^A	P_3^A	P_4^A
C_1^A	2827 ± 9270	433 ± 2392	9878 ± 19259	1260 ± 4960
C_2^A	1618 ± 9201	5522 ± 19308	7624 ± 13566	4271 ± 21538
C_3^A	2349 ± 12806	9473 ± 29293	17087 ± 28643	1375 ± 4627
C_4^A	5282 ± 29129	44291 ± 60309	58180 ± 131275	
C_5^A	3364 ± 12787	3279 ± 12733	40792 ± 56348	529 ± 3770

Figure 5 depicts the influence of the number of steps on the processor's working time of the transaction for various processes across all clients. We have considered the maximum number of steps as well as the average number of steps defined in a process and studied if there are any significant correlations. *We can see that as the maximum number of steps increases, the average working time decreases.* For a particular process, among clients where the maximum number of steps is the same, the average working time is directly proportional to the average number of steps in the process. For example, in Fig. 5(a) (also see Table 1), although both C_1^A and C_4^A have the same number of maximum steps (11), their average working times are 2827 and 5282 respectively (see Table 4). It can be concluded that, C_4^A takes relatively longer time because the average number of steps for this client is 5 as against 1.5 for client C_1^A .

Note that causal relations/correlations only suggest plausible factors for an observed behaviour. Valid conclusions can only be drawn in conjunction with process owners, while considering complete operational context. We believe this will have to be an iterative process, i.e., we discover insights from the data, present it to the process owners, incorporate their feedback, and if required repeat the analysis. The ability of a completely automated system for drawing confident conclusions largely depends on the goodness/quality of event data.

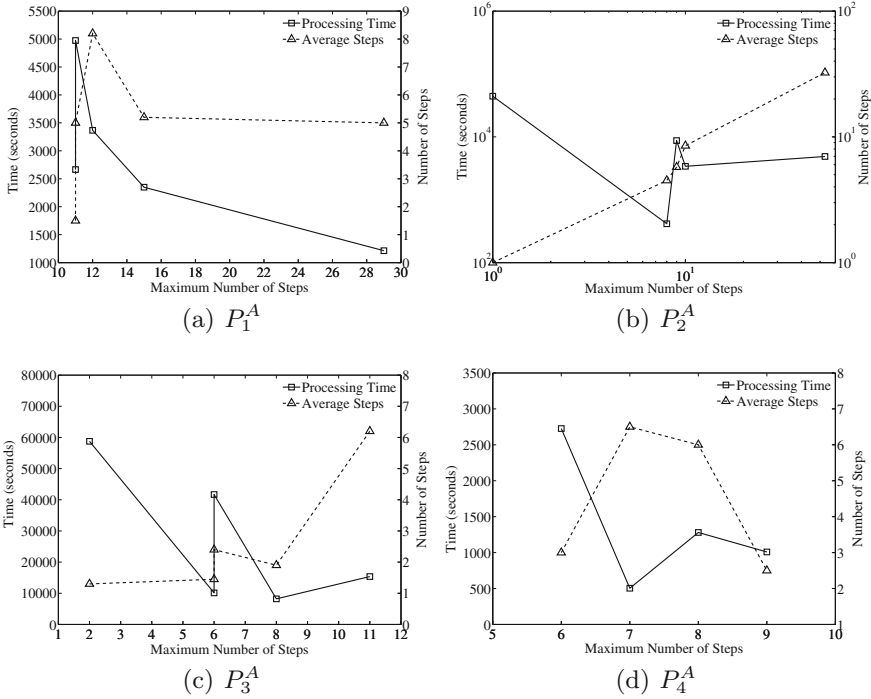


Fig. 5. Influence of the number of steps defined for a process on the processing time.

For example, apart from process data, one may also rely on information about resource activities each day, e.g., whether resources are working on other transactions with higher priority. Such completeness in event data logging is rarely observed. Therefore, for the study reported in the paper, we presented the uncovered causal relations to process owners, who concurred with the correlations, leading us to the stated conclusions.

In order to study the influence of resources on the variation in processing (working) time, we first filtered all infrequent transaction types and resources. We considered the transaction types that contribute to the top 95 percentile of transactions and resources who worked on at least 5% of transactions. Table 5 depicts the volume of transactions and the number of resources after this filtering. *We can see that the number of resources who regularly work on the process is much less when compared to all resources who worked on the process.* For example, for process P_1^A , there are 19 resources who worked on client C_1^A on all transactions but only 6 of them worked on at least 5% of transactions. When the volume of transactions arriving on certain days are high, in order to meet the SLAs, the organization deploys more resources to cater to the high volumes.

Figure 6 depicts the process model discovered using the heuristics miner for the client C_4^A and process P_1^A . We can see that in this model, the number of transactions with quality issues are 14 (0.35%) unlike the original model

Table 5. The number of transactions (#T), the number of processors (#R) deployed, and the average number of transactions per processor for different processes across different clients in domain A (considering only frequent transaction types and resources).

	P_1^A			P_2^A			P_3^A			P_4^A		
	#T	#R	avg.	#T	#R	avg.	#T	#R	avg.	#T	#R	avg.
C_1^A	1913	6	318.33	1189	2	594.50	504	11	45.82	1263	6	210.50
C_2^A	3711	5	742.20	328	7	46.86	252	6	42.00	987	6	164.50
C_3^A	1566	4	391.5	345	6	57.50	728	10	72.80	1109	6	184.83
C_4^A	4001	8	500.12	38	9	4.22	199	6	33.17			
C_5^A	1286	6	214.33	51	5	10.20	423	11	38.45	4617	5	923.40

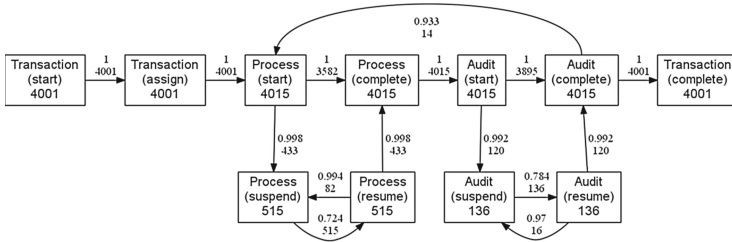


Fig. 6. Process model discovered using heuristics miner for client C_4^A and P_1^A considering only frequent transaction types and resources.

where there are 72 (1.13%) transactions with quality issues. *This indicates that resources who are deployed on a need basis being not so familiar with the process tend to do more mistakes.* Similarly, the number of times a processor suspends and resumes a transaction (10.82%) is lesser than the case when all resources are involved (16.01%).

Table 6 depicts the average and standard deviation of working time of the processors for the filtered log. We study if the experience and proficiency levels (in skills) of resources are potential causes for the variations in working times. Figure 7 depicts the relationship between the average working time of resources and their experience and skill proficiency for process P_1^A and two of the clients. We thus validate the intuitive results, that the *working times are inversely proportional to the experience of resources and directly proportional to their skill proficiency levels*, i.e., as resources gain experience, they tend to take less time. However, an experienced resource may take longer time than a less experienced resource if his/her proficiency level is lesser. We see an exceptional scenario in Fig. 7(b) where a resource with the maximum experience and proficiency level taking the longest time to execute this process. Upon further investigation, we uncovered that this resource, although processing large volumes of transactions for this process, is also involved in processing transactions related to another process. The context of the other process (demand, SLAs etc.) might require

Table 6. The average and standard deviation of working times (in seconds) taken by processors for different processes across different clients in domain A (considering only frequent transaction types and resources).

	P_1^A	P_2^A	P_3^A	P_4^A
C_1^A	2665 ± 9209	415 ± 2514	10125 ± 18704	1009 ± 2822
C_2^A	1213 ± 9015	4871 ± 18078	8214 ± 14503	2726 ± 15296
C_3^A	2349 ± 13268	8722 ± 27687	15373 ± 22996	1280 ± 4941
C_4^A	4974 ± 31383	44291 ± 60309	58760 ± 131807	
C_5^A	3367 ± 13333	3400 ± 13099	41692 ± 56969	504 ± 2236

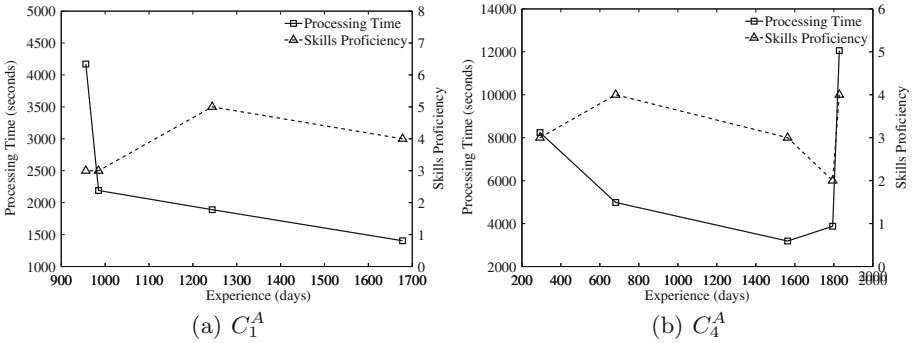


Fig. 7. Relationship between experience and proficiency in skills of resources and their working/processing time

him to be more involved in the execution of that process than this one, thereby resulting in longer working times for this process.

We have analyzed the event data pertaining to various client-process combinations in another domain and the basic observations discussed so far manifested also in the other domain. To summarize, our key findings are:

- the granularity at which process steps are defined influences the processing time; fine-granular steps help processors comprehend the task easily and thereby lead to efficient execution
- a resource’s experience and skill proficiency significantly impact the processing time
- adhoc resources deployed for executing a process to cater to large volume of transactions (on certain days) tend to make mistakes while executing the task (especially, for complex processes) impacting both the quality as well as processing time.

Based on our analysis, we prescribe that (i) processes should be defined in detail to help processors execute the process efficiently; (ii) stable resources should be deployed as far as possible to execute a process. Even if resources are to be

deployed temporarily to cater to the demand, it is advisable to have a limited pool of resources who are familiar with the process; (iii) resource experience and skill proficiency should both be considered when forming team compositions. These prescriptions are envisioned to be part of the recommendation engine after discussing with the business stakeholders.

6 Conclusions

Process improvement efforts based on insights obtained from the analysis of operational event data has primarily been applied on individual processes separately. Analysis of event logs pertaining to a similar process across different clients provides valuable insights on the operations of an organization. In this paper, we provided a framework that enables cross-client analysis and presented results on applying this framework on a real-life case study from a services organization. We uncovered that detailed description of process definitions, balanced workload, stable, proficient and experienced resources all lead towards operational excellence. Organizations can incorporate these learnings to aspects such as team formation (skill/experience dependent), training, workload distribution, task description etc. In addition to providing insights from cross-client learnings, we envision that our framework can assist service organizations in determining the health of client-process combinations and provide client-process benchmarks. As future work we would like to build a recommendation engine that provides prescriptions for operational excellence augmented with a simulation engine that corroborates with *what-if* analysis of the prescriptions.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery Conformance and Enhancement of Business Processes. Springer New York Inc., New York (2011)
2. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisc. Rev. Data Min. Knowl. Discov.* **2**(2), 182–192 (2012)
3. van der Aalst, W.M.P., de Beer, H.T., van Dongen, B.F.: Process mining and verification of properties: an approach based on temporal logic. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 130–147. Springer, Heidelberg (2005)
4. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Conformance checking using cost-based fitness analysis. In: Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 55–64 (2011)
5. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards cross-organizational process mining in collections of process models and their executions. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBP, vol. 100, pp. 2–13. Springer, Heidelberg (2012)
6. CoSeLoG (2014). http://www.win.tue.nl/coselog/wiki/project#the_project
7. van Dongen, B.F., van der Aalst, W.M.P.: A meta model for process mining data. In: Proceedings of the CAiSE Workshops (EMOI-INTEROP Workshop), vol. 2, pp. 309–320 (2005)

8. Ferreira, D.R., Vasilyev, E.: Using logical decision trees to discover the cause of process delays from event logs. *Comput. Ind.* **70**, 194–207 (2015)
9. Günther, C.W.: XES Standard Definition (2009). www.xes-standard.org
10. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) *BPM 2011*. LNCS, vol. 6896, pp. 132–147. Springer, Heidelberg (2011)
11. Nguyen, H., Dumas, M., La Rosa, M., Maggi, F.M., Suriadi, S.: Mining business process deviance: a quest for accuracy. In: Meersman, R., Panetto, H., Dillon, T., Missikoff, M., Liu, L., Pastor, O., Cuzzocrea, A., Sellis, T. (eds.) *OTM 2014*. LNCS, vol. 8841, pp. 436–445. Springer, Heidelberg (2014)
12. Pika, A., van der Aalst, W.M.P., Fidge, C.J., ter Hofstede, A.H.M., Wynn, M.T.: Predicting deadline transgressions using event logs. In: La Rosa, M., Soffer, P. (eds.) *BPM Workshops 2012*. LNBIP, vol. 132, pp. 211–216. Springer, Heidelberg (2013)
13. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
14. Sindhgatta, R., Dasgupta, G.B., Ghose, A.: Analysis of operational data for expertise aware staffing. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) *BPM 2014*. LNCS, vol. 8659, pp. 317–332. Springer, Heidelberg (2014)
15. Suriadi, S., Ouyang, C., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Root cause analysis with enriched process logs. In: La Rosa, M., Soffer, P. (eds.) *BPM Workshops 2012*. LNBIP, vol. 132, pp. 174–186. Springer, Heidelberg (2013)
16. Verbeek, H.M.W., Buijs, J.C.A.M., Dongen, B.F.V., van der Aalst, W.M.P.: ProM 6: the process mining toolkit. In: *CEUR Workshop Proceedings*, vol. 615, pp. 34–39 (2010)
17. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data using little thumb. *Integr. Comput.-Aided Eng.* **10**(2), 151–162 (2003)