Business Process Performance Mining with Staged Process Flows

Hoang Nguyen[1], Marlon Dumas[2], Arthur H.M. ter Hofstede[1], Marcello La Rosa[1,3], and Fabrizio Maria Maggi[2]

[1] Queensland University of Technology, Australia
huanghuy.nguyen@hdr.qut.edu.au,
{a.terhofstede,m.larosa}@qut.edu.au
[2] University of Tartu, Estonia
{marlon.dumas,f.m.maggi}@ut.ee
[3] NICTA Queensland, Australia

Abstract. Existing process mining techniques provide summary views of the overall process performance over a period of time, allowing analysts to identify bottlenecks and associated performance issues. However, these tools are not designed to help analysts understand how bottlenecks form and dissolve over time nor how the formation and dissolution of bottlenecks – and associated fluctuations in demand and capacity – affect the overall process performance. This paper presents an approach to analyze the evolution of process performance via a notion of Staged Process Flow (SPF). An SPF abstracts a business process as a series of queues corresponding to stages. The paper defines a number of stage characteristics and visualizations that collectively allow process performance evolution to be analyzed from multiple perspectives. The approach has been implemented in the ProM process mining framework. The paper demonstrates the advantages of the SPF approach over state-of-the-art process performance mining tools using a real-life event log of a Dutch bank.

1 Introduction

Process mining is a family of techniques designed to extract insights from business process event logs [1]. *Process Performance Mining* (PPM) is a subset of process mining techniques concerned with the analysis of processes with respect to performance dimensions, chiefly *time* (how fast a process is executed); *cost* (how much a process execution costs); *quality* (how well the process meets customer requirements and expectations); and *flexibility* (how rapidly can a process adjusts to changes in the environment) [2].

Along the time and flexibility dimensions, one recurrent analysis task is to understand how the temporal performance of a process evolves over a given period of time – also known as *flow performance* analysis in lean management [3]. For example, a bank manager may wish to know how the waiting times in a loan application process have evolved over the past month in order to adjust the resource allocation policies so as to minimize the effects of bottlenecks.

Existing PPM techniques are not designed to address such flow performance questions. Instead, these techniques focus on analyzing process performance in a "snapshot" manner, by taking as input an event log recorded during a period of time and extracting aggregate measures such as mean waiting time, processing time or cycle time of the process and its activities. For example, both the Performance Analysis plugins of ProM [4] and Disco [5] calculate aggregate performance measures (e.g. mean waiting time) over the entire period covered by an event log and display these measures by color-coding

the elements of a process model. These tools can also produce animations of the flow of cases along a process model over time. However, extracting flow performance insights from these animations requires close and continuous attention from the analyst in order to detect visual cues of performance trends, bottleneck formation and dissolution, and phase transitions in the process' performance. In other words, animation techniques allow analysts to get a broad picture of performance issues, but not to precisely quantify the evolution of process performance over time.

In this setting, this paper presents a PPM approach designed to provide a precise and quantifiable picture of flow performance. The approach relies on an abstraction of business processes called *Staged Process Flow* (SPF). An SPF breaks down a process into a series of queues corresponding to stages. Each stage is associated with a number of performance characteristics that are computed at each time point in an observation window. The evolution of these characteristics is then plotted via several visualization techniques that collectively allow flow performance to be analyzed from multiple perspectives in order to address the following questions:

Q1. How does the overall process performance evolve over time?
Q2. How does the formation and dissolution of bottlenecks affect the overall process performance?
Q3. How do changes in demand and capacity affect overall process performance?

The rest of this paper is organized as follows. Section 2 reviews existing PPM techniques with respect to the problem of flow performance analysis. Section 3 describes the SPF concept and associated characteristics and visualizations. Section 4 discusses an evaluation of the approach based on a real-life log. Finally, Section 5 summarizes the contributions and outlines future work directions.

2 Related Work

Existing PPM tools support the analysis of entire processes or activities thereof with respect to typical performance measures such as cycle time, processing time and waiting time. Some PPM tools display the distribution of performance measures in the form of dashboards (e.g. bar charts) alongside aggregate statistics (e.g. mean and median) [5]. Others overlay the performance measures on top of a process model, for example by replaying the log on the process model [4, 6] and calculating aggregate performance measures for each element in the process model during replay. Techniques for enhancing the quality of log replaying based on clustering techniques have been proposed [7, 8]. All these techniques are designed to summarize the performance of the process over the entire time period covered by the event log. They can pinpoint bottlenecks, resource underutilization and other performance issues observed across said time period. However, they do not allow one to analyze how those bottlenecks form and dissolve, and more generally, how the performance of the process varies over time.

There is a range of techniques to extract and analyze process performance characteristics (incl. performance measures) from event logs. For example, De Leoni et al. [9] propose a framework to extract process performance characteristics from event logs and to correlate them in order to discriminate for example between the performance of cases that lead to "positive" outcomes versus "negative" outcomes. Meanwhile, Pika et al. [10] propose a framework to extract performance characteristics along the resource perspective. These proposals however are not designed to provide insights into the evolution of process performance over time.

A related technique supported by contemporary PPM tools is log animation. Log animation displays in a movie-like fashion how cases circulate through the process

model over time [11, 7, 12]. However, extracting flow performance insights from these animations requires the analyst to: (i) manually look for visual cues in the animation that indicate trends, phase transitions or bottlenecks in the process' performance; and (ii) run additional queries to locate and quantify the observed performance phenomena.

Process performance has also been approached from the perspective of queuing theory. Senderovich et al. [13] propose a method to discover characteristics of "work queues" from event logs at the level of an entire process or of individual activities. Meanwhile, Smet [14] proposes a method to discover collections of queues from event logs. This latter method discovers queues by grouping resources and activities into clusters based on cohesion metrics. The queuing models produced by the above methods are used for prediction (e.g. of waiting times) rather than performance analysis, As such these methods are only marginally related to the problem of flow performance analysis.

The concept of SPF presented in this paper is inspired by flow performance analysis techniques from the fields of lean management and agile software engineering. The idea of decomposing the process into stages and analyzing flow metrics at each stage can be found in various embodiments in contemporary lean and agile management tools, e.g. Kanban Flow[4], Kanban Tool[5], Scrumwise[6] and ActionableAgile[7]. The concept of SPF formalized in this paper in the context of business process event logs, provides a generic framework that brings together flow performance analysis techniques found across these tools.

3 Approach

In this section, we introduce the concept of SPF and its formalization before describing our SPF-based approach to process performance mining.

3.1 SPF overview

An SPF is a series of *stages*, each one grouping together multiple activities. A stage is modeled as a *queuing system*, where the queuing items are cases and the service facility is the set of resources available to handle cases in the stage in question. Each stage has an *arrival flow* via which new cases arrive to the stage in question and a *departure flow* via which cases depart. In addition, a stage may have *exit flows*, capturing the fact that a case may leave the process abnormally after being serviced at a stage.

For illustration, we use a loan origination process at a Dutch bank that was the subject of the BPI Challenge 2012 log [15]. As depicted in the SPF in Fig. 1, a case in this process is a loan application that goes through four *stages*: Pre-Assess, Assess, Negotiate, and Validate. In the "Pre-Assess" stage, the bank checks the completeness of the loan application and requests the customer to provide sufficient documents before their application can proceed to the next stage. In the "Assess" stage, the bank checks the eligibility of the loan application. Next, in the "Negotiate" stage, the bank and the customer discuss the terms and conditions of the loan until it is ready for validation. Finally, in the "Validate" stage, a bank controller makes a review and decides to approve or reject the loan application. At the end of any stage, a loan application can either be declined by the bank or canceled by the customer, which leads to interrupting the process at that point. Note that each stage has two exit flows, corresponding to loan applications that are declined or cancelled.

[4] http://kanbanflow.com
[5] http://kanbantool.com
[6] http://www.scrumwise.com
[7] http://www.actionableagile.com

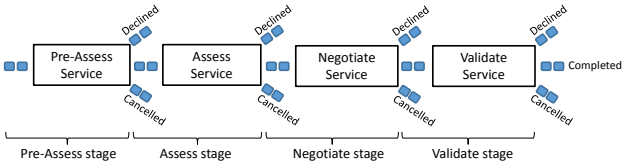


Fig. 1. SPF model of a loan origination process

Flow performance in an SPF is determined by a set of *characteristics* capturing the interplay between the arrival flow on the one hand and the departure and exit flows on the other. One such characteristic is the *Cases In Progress* (in reference to "Work-in-Progress"), that is, the set of cases found in a stage at a given point in time. Another characteristic is the *Time in Stage*: the time between the arrival and the departure/exit of a case for a given stage. Each case spends a certain amount of time waiting in a stage, and another amount of time being processed in that stage. *Flow efficiency* is the ratio between the processing time of a case in a stage and the Time in Stage. Below we formally define how an SPF and its characteristics are extracted from an event log.

3.2 SPF formalization

An event log is the starting point of any process mining task. Fig. 2 shows an event log of a loan origination process. An event log consists of a set of *cases*, where a case is a uniquely identified instance of a process. For example, the loan application identified by code *L344* is a case. Each case consists of a sequence of *events*. An event is the most granular element of a log and is characterised by a set of attributes such as *activity*, *resource* (the entity that performed the activity associated with the event, which can be human or non-human), and *timestamp* (the moment when the event occurred). *Event type* represents the association between an event and its activity's lifecycle, such as "schedule", "start", and "complete". In this paper we assume that "start" and "complete" are the only event types associated with activities. In addition we assume that for every activity in the event log there is both a corresponding start event and a complete event (i.e. we only consider completed cases). Activities are identified by a globally (i.e. across cases) unique identifier. This allows us to compute the duration of activities.

Case ID	Case Status	Event ID	Stage	Event Type	Timestamp	Activity Name	Resource
C1	Declined	001	Pre-Assess	start	05.10.2011 03:36:15	Complete application	Rob
				complete	05.10.2011 04:05:15	Complete application	Rob
		002	Pre-Assess	start	06.10.2011 03:36:15	Complete application	Sara
				complete	06.10.2011 04:05:15	Complete application	Sara
C2	Completed	001	Pre-Assess	start	08.10.2011 09:30:15	Complete application	Tim
				complete	08.10.2011 10:00:15	Complete application	Tim
		002	Assess	start	10.10.2011 09:36:15	Check application	Tim
				complete	10.10.2011 09:45:00	Check application	Tim
		003	Negotiate	start	11.10.2011 15:05:00	Follow up offer	Tim
				complete	11.10.2011 16:05:00	Follow up offer	Tim
		004	Validate	start	14.10.2011 10:00:00	Validate application	Michael
				complete	14.10.2011 10:30:15	Validate application	Michael

Fig. 2. Example event log for a loan origination process

Formally, an event log EL is a tuple $(E, ET, A, R, C, time, act, type, res, case)$, where E is a set of events, $ET = \{start, complete\}$ is the set of event types, A is a set of activity identifiers (AID), R is a set of resources, C is a set of cases, $time : E \rightarrow \mathbb{R}_0^+$ is a function that assigns a timestamp to an event, $act : E \rightarrow A$ is a function that assigns an activity identifier (AID) to an event, $type : E \rightarrow ET$ is a function that assigns an

event type to an event, $res : E \rightarrow R$ is a function that assigns a resource to an event, and $case : E \rightarrow C$ relates an event to a case. We write $e \lesssim_E e'$ iff $time(e) \leq time(e')$.

In our model, events are associated with *stages*. For example, a particular "Complete application" event occurs at the "Assess" stage of a loan application. A completed case is one that passed all stages and has a "complete" status, otherwise, the case is considered to have exited the process prematurely and will have the status "incomplete".

Stage-based enhancement In our approach, an event log must firstly be enhanced with stage information. A *stage-based enhancement* SE of an event log EL is defined as a tuple $(S, CS, <_S, stage, status)$, where S is a set of stages, $CS = \{complete, incomplete\}$ a set of cases statuses, $<_S \subseteq S \times S$ a strict total order over S (with $\leqslant_S$ the corresponding total order), $stage : E \rightarrow S$ assigns stages to events, and $status$ assigns statuses to cases. For convenience we write $E^{c,s} = \{e \in E \mid case(e) = c \wedge stage(e) = s\}$ to denote the set of all events of case c that occurred in stage s. $E^{start} = \{e \in E \mid type(e) = start\}$ is the set of all start events.

While there can be a number of ways to arrive at a stage-based enhancement of an event log, there are a number of rules that need to be satisfied. First of all, if a case covers a stage s, i.e. there is at least one event belonging to that stage, there must be events associated with all stages preceding s in that case:

$$\forall c \in C \forall e \in E[E^{c,s} \neq \varnothing \Rightarrow \forall s' \in S[s' <_S stage(e) \Rightarrow E^{c,s'} \neq \varnothing]].$$

The stages covered by a case must observe the defined order $<_S$ over S:

$$\forall e, e' \in E[(case(e) = case(e') \wedge e \lesssim_E e') \Rightarrow stage(e) \leqslant_S stage(e')].$$

Events related to an AID must belong to the same stage:

$$\forall e, e' \in E[act(e) = act(e') \Rightarrow stage(e) = stage(e')].$$

If a case has a complete status, it should have gone through all the stages:

$$\forall c \in C[status(c) = complete \Rightarrow \forall s \in S \; \exists e \in E[case(e) = c \wedge stage(e) = s]].$$

SPF characteristics The start of stage s in case c, $\mathbf{T}_{AR}(\mathbf{c}, \mathbf{s})$, is defined as $\min_{e \in E^{c,s}} time(e)$ if $E^{c,s} \neq \varnothing$ and is undefined ($\perp$) otherwise. Similarly, the end of a stage s in case c, $\mathbf{T}_{DP}(\mathbf{c}, \mathbf{s})$, is $\max_{e \in E^{c,s}} time(e)$ if $E^{c,s} \neq \varnothing$ and is undefined ($\perp$) otherwise. For all timestamps t neither $t < \perp$ nor $t > \perp$ holds. The last stage of case c is s, $laststage(c, s)$, iff $\neg \exists s' \in S \exists e \in E[s <_S s' \wedge case(e) = c \wedge stage(e) = s']$.

The set $\mathbf{C}_{AR}(\mathbf{s}, \mathbf{t})$ consists of all cases that have reached stage s on or before t, i.e. $\mathbf{C}_{AR}(\mathbf{s}, \mathbf{t}) \triangleq \{c \in C \mid \exists e \in E^{c,s}[time(e) \leq t]\}$. Similarly, the set $\mathbf{C}_{DP}(\mathbf{s}, \mathbf{t})$ consists of all cases that have gone beyond stage s on or before time t, i.e. $\mathbf{C}_{DP}(\mathbf{s}, \mathbf{t}) \triangleq \{c \in C \mid \forall e \in E^{c,s}[time(e) \leq t]\}$. Looking at the inflows and outflows, the set $\mathbf{C}_{IN}(\mathbf{s}, \mathbf{t_1}, \mathbf{t_2})$ consists of those cases that have events occurring in stage s during interval (t_1, t_2), denoted $\mathbf{C}_{IN}(\mathbf{s}, \mathbf{t_1}, \mathbf{t_2}) \triangleq \{c \in C \mid \exists e \in E^{c,s}[t_1 \leq time(e) \wedge time(e) \leq t_2]\}$, while set $\mathbf{C}_{EX}(\mathbf{s}, \mathbf{t})$ consists of those cases that have completed stage s on or before time t, have not gone beyond stage s, and are considered to be incomplete: $\mathbf{C}_{EX}(\mathbf{s}, \mathbf{t}) \triangleq \{c \in C \mid \forall e \in E^{c,s}[time(e) \leq t] \wedge laststage(c, s) \wedge status(c) \neq complete\}$. Let $a \in A$ be an AID, $e_s \in E$ an event for which $act(e_s) = a$ and $type(e_s) = start$, $e_c \in E$ an event for which $act(e_c) = a$ and $type(e_c) = complete$, then the duration of AID a is defined as $dur(a) = time(e_c) - time(e_s)$ (we assume that this is always greater than zero) and the duration of this AID during a closed time interval $[t_1, t_2]$, denoted $dur(a, t_1, t_2)$, is defined as $[t_1, t_2] \cap [time(e_s), time(e_c)]$.

The duration of a case c at a stage s within interval (t_1, t_2), written $dur(c, s, t_1, t_2)$, is defined as $[t_1, t_2] \cap [\mathbf{T}_{AR}(\mathbf{c}, \mathbf{s}), \mathbf{T}_{DP}(\mathbf{c}, \mathbf{s})]$. This notion is defined when $E^{c,s} \neq \varnothing$.

The *Arrival/Departure/Exit Rate* X is the average number of cases arriving at/departing from/exiting after a stage per unit of time at a given point in time.

$$X(s, t, \Delta) \triangleq \frac{C_X(s, t) - C_X(s, t - \Delta)}{\Delta}.$$

It is required that $\Delta > 0$ (here and elsewhere) and $t - \Delta$ does not occur before all case start times in the log, i.e. $\exists e \in E[time(e) \leq (t - \Delta)]$.

Cases in Progress is the number of cases present at a stage at a time point.

$$CIP(s, t, \Delta) \triangleq |C_{AR}(s, t, \Delta)| - |C_{DP}(s, t, \Delta)|.$$

The *Time in Stage* for a point in time t and a stage s is the minimal duration that one needs to wait to see the number of departing cases from s equal the number of cases that arrived in stage s on or before time t. Formally, let t_o be the minimal timestamp greater than t for which $|C_{DP}(s, t_o, \Delta)| = |C_{AR}(s, t, \Delta)|$, then $TIS(s, t, \Delta) = t_o - t$. $TIS(s, t, \Delta)$ is undefined if no such t_o exists.

Finally, the *Flow Efficiency* FE of a stage s during an interval $[t - \Delta, t]$ is the sum of all activity durations for all cases in that stage in that interval divided by the sum of all case durations for that stage in said interval:

$$FE(s, t, \Delta) \triangleq \frac{\sum_{e \in E^{start}, stage(e) = s} dur(act(e), t - \Delta, t)}{\sum_{e \in E^{start}} dur(case(e), s, t - \Delta, t)}.$$

Flow efficiency can be computed for logs where all AIDs have both a start and a complete event. Also, at least one case should have an AID that has a non-zero duration in the interval $[t - \Delta, t]$ in stage s in that case, to avoid the denominator evaluating to zero.

3.3 SPF-based performance mining approach

Our approach to process performance mining follows three steps: i) construct the SPF from an event log; ii) measure the flow performance using the SPF; and iii) visualize the process performance and SPF characteristics for user consumption.

Constructing the SPF from an event log To construct the SPF, we first enhance the event log with stage information. This is done via preprocessing, which consists in adding two stage-related attributes: a "stage" attribute for each event, indicating which stage it belongs to, and a "status" attribute to the case, indicating if the case is complete.

Given a stage-enhanced event log, the SPF is then constructed via the following procedure. First, a stage-based timetable is created. Fig. 4 shows an example timetable created from the log of the loan origination process of Fig. 2, where S1, S2, S3 and S4 stand for Pre-Assess, Assess, Negotiate and Validate stage, respectively. Every row of the table is the stage-based timetable of a case. The start and end timestamps of a stage on one row are the timestamp of the first

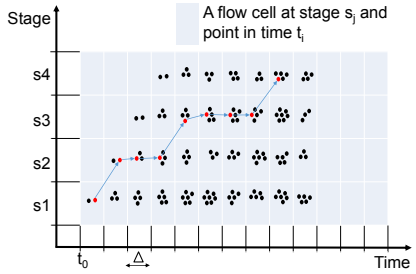


Fig. 3. Example flow plane with a case trajectory

and last event of that stage, for a given case, e.g. for C1. Next, the *case trajectory* is computed as follows. The entire timeline of the log is divided into even time intervals Δ. The stages and time create a two-dimensional space called *flow plane* of the log (see Fig. 3). In this space, a *flow cell* is located at a stage and a point in time $t_i = t_o + i\Delta$ ($i = 0 \ldots n$), with t_o being the starting time of the log. The trajectory of a case on the flow plane is computed by checking the stage-based timetable of the case to determine which flow cells in the plane are covered, whereby a cell is covered by a stage if its time (t_i) is within the start and end timestamp of that stage. In other words, the trajectory of a case consists of flow cells where the case resides during its lifecycle. For example, Fig. 3 shows the trajectory of case C2 from Fig. 4. By computing the trajectory of all cases, we can construct the SPF.

Case ID	S1_start	S1_end	S2_start	S2_end	S3_start	S3_end	S4_start	S4_end
C1	05.10.2011 03:36:15	06.10.2011 04:05:15						
C2	08.10.2011 09:30:15	08.10.2011 10:00:15	10.10.2011 09:36:15	10.10.2011 09:45:00	11.10.2011 15:05:00	11.10.2011 16:05:00	14.10.2011 10:00:00	14.10.2011 10:30:15

Fig. 4. Stage-based timetable of the loan origination process of Fig. 2 (S1=Pre-Assess, S2=Assess, S3=Negotiate, S4=Validate)

Measuring flow performance SPF characteristics are computed first at every flow cell. Then, the flow performance is measured at the stage level and system level (the whole SPF). Let φ be a SPF characteristic such as Arrival Rate or Departure Rate. The value of φ at a stage s_j and a point in time $t_o + i\Delta$ (i=0...n), denoted as $\varphi(s_j, t_0 + i\Delta, \Delta)$, can be computed using the formulae of the SPF characteristics presented in Section 3.2.

Let Γ be a statistical function, e.g. mean or median. The characteristic φ at stage $s_j \in S$ for a time interval (t_1, t_2), denoted as $\varphi(s_j, t_1, t_2)$, is defined as $\Gamma(\varphi(s_j, t_0 + n\Delta, \Delta), \ldots, \varphi(s_j, t_0 + m\Delta, \Delta))$, where $n < m$ and $(t_o + n\Delta, t_o + m\Delta)$ is the maximal timespan contained within (t_1, t_2).

The flow performance at the system level is based on that at the stage level but computed differently for some characteristics. The Arrival Rate of the system is the Arrival Rate at the first stage. The Departure Rate of the system is the Departure Rate at the last stage. The other characteristics at the system level for a period (t_1, t_2), denoted as $\varphi(S, t_1, t_2)$, are defined as $\Gamma(\varphi(s_1, t_1, t_2), \ldots, \varphi(s_m, t_1, t_2)$, where m is the total number of stages in the system, i.e. $m = |S|$.

Investigating the SPF and its characteristics Based on the above formalization, we provide three visualizations to support the analysis of SPF characteristics at different levels of abstraction and periods of time.

Cumulative Flow Diagram (CFD): A CFDs is an area graph used in queueing theory [16] to visualize the evolution of flow performance over time. Figure 5 depicts an example CFD showing how some SPF characteristics are related to the geometry of the diagram. In our case, each area, encoded with a different color, represents the number of cases queuing for a given process stage (arrival flow), being worked in that stage (stage flow) or existing from that stage (exit flow). The CFD offers a number of features that are particularly suitable for examining

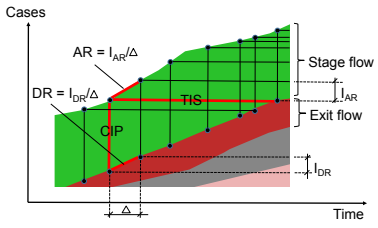


Fig. 5. CFD structure

the SPF and the flow performance. First, one can identify process trends by observing the development of the different SPF flows over time, individually or aggregately. An an example, an increase of both cases in progress (CIP) and time in stage (TIS) in a given stage, denotes the formation of a bottleneck. The CFD can show this easily through widening areas on a stage flow. Moreover, by drilling up and down the CFD to view the SPF at different levels of detail (e.g. hourly, daily, weekly or monthly), one may spot the occurrence of any recurring process behavior (patterns) during a given timeframe, e.g. there is some activity at nighttime every Friday, which may signal the fact that some cases must be moved to the next stage by the end of the working week. Finally, as a CFD can integrate multiple SPF characteristics into a single diagram, it can be helpful to detect correlations between these metrics, such as between arrival rate and departure rate, or between departure rates across different stages.

Performance Summary Table (PST): the PST (Fig. 6) provides a quick and exact measurement of the flow performance in figures, at the stage and system levels. It also allows one to measure the flow for any time interval of the log.

From Sat, 1 Oct 2011 00:38:44 +0200 To Wed, 22 Feb 2012 18:24:21 +0100 (144 days) (Mean/Median)

	AR(cases/day)	DR(cases/day)	ER(cases/day)	CIP(cases)	TIS(hours)	FE(%)
System	84.73 / 72.00	18.62 / 0.00	65.72 / 24.00	702.26 / 723.00	550.33 / 546.00	2.92
s1-queue	84.73 / 72.00	84.73 / 72.00		3.53 / 3.00	0.69 / 0.00	
s1-service	84.73 / 72.00	84.69 / 48.00	37.10 / 24.00	12.19 / 10.00	4.13 / 2.00	9.25 / 1.35
s2-queue	47.59 / 24.00	47.55 / 24.00		4.44 / 3.00	2.69 / 1.00	
s2-service	47.55 / 24.00	46.38 / 0.00	14.05 / 0.00	132.53 / 136.00	67.58 / 68.00	0.86 / 0.63
s3-queue	32.33 / 0.00	31.49 / 0.00		96.98 / 97.00	72.64 / 71.00	
s3-service	31.49 / 0.00	28.39 / 0.00	9.22 / 0.00	362.85 / 379.00	285.25 / 296.00	0.23 / 0.19
s4-queue	19.17 / 0.00	19.03 / 0.00		48.08 / 52.00	61.40 / 58.00	
s4-service	19.03 / 0.00	18.62 / 0.00	5.35 / 0.00	41.66 / 43.00	55.95 / 50.00	1.32 / 0.94

Fig. 6. Performance Summary Table (AR=Arrival Rate, DR=Departure Rate, ER=Exit Rate, CIP=Cases in Progress, TIS=Time in Stage, FE=Flow Efficiency)

Time Series Charts (TSCs): As most SPF characteristics are time-dependent, TSCs (Fig. 7) can be used to investigate the evolution of SPF stage characteristics over time, such as viewing the development of arrival rate, the difference between departure and arrival rate at different intervals, or the formation of a bottleneck over time. Fig. 7 gives a multiple-series TSC showing the evolution of various SPF characteristics over time.

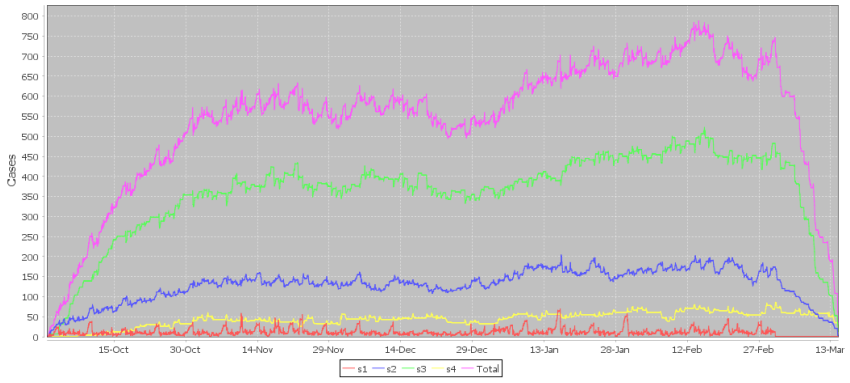


Fig. 7. Time Series Chart of various SPF characteristics

SPF Comparison: from the original SPF computed for the whole event log, multiple SPFs can be produced via filtering, e.g. by case attributes such as case status, amount

and case time. We can then investigate the characteristics of these SPFs and compare the filtered SPF with its complete counterpart, for answering a question such as which SPF accounts for the most significant part of the complete SPF.

4 Evaluation

We implemented our approach as a ProM plugin, namely the "Performance Mining With Staged Process Flows" plugin, as well as a standalone Java application[8]. In the following, we use this implementation to answer the questions raised in Section 1 against the BPI Challenge 2012 log [15], and compare the results with those obtained from two state-of-the-art PPM tools. For space reasons, the results of a second evaluation, using the BPI Challenge 2013 log [17], are provided in a technical report [18], though they are in line with those reported in this paper.

The BPI Challenge 2012 log records cases of a loan origination process at a Dutch bank (see Section 3 for a description). It contains 13,087 loan applications with a total of 193,208 events occurring from 1 Oct 2011 to 15 Mar 2012. Every case must pass four stages. The completion of each stage is marked by a special event such as A_PREACCEPTED, A_ACCEPTED, and A_ACTIVATED. We preprocessed this log to enhance it with stage information, including adding a "stage" attribute for events and a "status" attribute for cases.

The PPM tools evaluated are the "Performance Analysis with Petri Net" plugin of ProM 5.2 [4] (PEP for short), and Fluxicon's Disco [5]. The tool requires a Petri net discovered from an event log as input. The net can be obtained by using any of the available discovery algorithms of ProM that either directly discovers a Petri net or whose result can be converted into a Petri net, such as the Heuristics Miner. PEP can be run to internally replay the log on the Petri net, in order to align the log with the model, compute time-related performance information and overlay it to the model. Specifically, processing time is assigned to Petri net transitions (capturing process activities) while waiting time is assigned to places (capturing states). Arrival rates for these elements are also provided. Moreover, places are color-coded based on the length of the waiting time (blue for short waits, yellow for medium and red for long). The thresholds for the colors can be set automatically or manually by the user. The tool also provides overall performance measures such as arrival rate and statistics on cycle time.[9]

Similar to PEP, Disco's performance measurements are mainly based on a process model. The tool takes an event log as input and discovers a Fuzzy net, which provides an abstract representation of the process behavior, by showing the process activities and paths connecting these activities. This model is enhanced with frequency information and statistics on performance measures at the level of individual process activities (processing time) and paths (waiting time). The complexity of the discovered model can be adjusted based on case frequency, in order to obtain a simpler process model that abstracts away infrequent cases. Different types of filters besides frequency can be used to create model projections which can be used to compare process variants on the basis of their performance, e.g. focusing on all cases that have a duration or a number of events within a given range. In addition, Disco can replay the log on the discovered model.

For each question, we evaluated each tool along the quality dimensions of *ease of use* and *usefulness*, widely used in technology acceptance models [19]. In our context,

[8] Available from `http://promtools.org` (ProM) and `http://apromore.org/platform/tools` (standlone Java application)

[9] The "Replay a Log on Petri Net for Performance/Conformance Analysis" plugin of ProM 6 works in a similar way to PEP, though it provides less performance information.

ease of use refers to the effort required from the user to retrieve and to interpret data in order to answer a given question. Usefulness on the other hand refers to the extent the tool provides data that allows the user to answer the question in a precise (i.e. quantitatively) and informative manner. Below we evaluate the three tools for each question.

4.1 Loan Origination Process

Q1: How does the overall process performance evolve over time?

SPF The evolution of the process is depicted on the CFD (Fig. 8). The shape of the CFD reflects the development of the process at each stage. The characteristics, such as arrival rate (AR) and cases in process (CIP), can be seen at any point in time as a tooltip. The CFD can be zoomed in to investigate patterns of evolution at different intervals (e.g. weekly, daily and hourly). The evolution can also be viewed on the plot of flow efficiency over time. The PST (Fig. 6) provides a summary of the flow performance at any time interval. From these visualizations, we can draw the following observations:

- The process has a stable trend indicated through the even height of service bands shown in Fig. 8 (bands named as si-Service). Further evidence is provided by the average arrival and departure rates, which are comparable at each stage in Fig. 6, and by the fact that there is little variation between the average mean and median value of CIP and TIS.
- There are strong exit flows throughout the period from s1 (strongest) to s4 (bands named as si-Exit on the CFD). Fig. 9 shows a distribution of types of exits at every stage. Apparently, these exit flows contribute to keeping the arrival of cases at each stage on a par with their departure.
- Zooming into an hourly level of detail, we can observe a constraint of a two-hour window for the delivery at stage s3, as shown in Fig. 13.b.
- The CFD and PST show that the waiting queue is negligible at stage s1 but starts to emerge at stage s2 and becomes considerable at stages s3 and s4, meaning that the process has slower response in the later stages.
- The process has very low flow efficiency (3%), meaning more than nearly 97% of time a case stays idle. The problem seems to be with the customer information as the process involves intensive interaction with customers.

As shown above, the SPF proposes an easy way to understand how the overall process performance evolves over time. The output is easy to interpret, as it is based on visual cues and performance measures; precise, as it is supported by numeric measurements; and most importantly, it leads to various insightful observations.

PEP An excerpt of the Petri net enhanced with performance information provided by PEP is shown in Fig. 10. This model was obtained by first discovering a Heuristics net and then converting it to a Petri net. However, in order to obtain a model that is easy to interpret, we had to incrementally filter the log, as the first model discovered was a spaghetti-like model too dense to understand. Eventually, we ended up retaining only those events that mark the end of each stage in the log (i.e. the "gate" events), in a similar vein to our approach. A drawback of this operation is that the fitness of the model decreases as some traces of the log can no longer be replayed on the log. As a result, the performance measures provided by the tool are only approximate, as they only refer to those traces that perfectly fit the model.

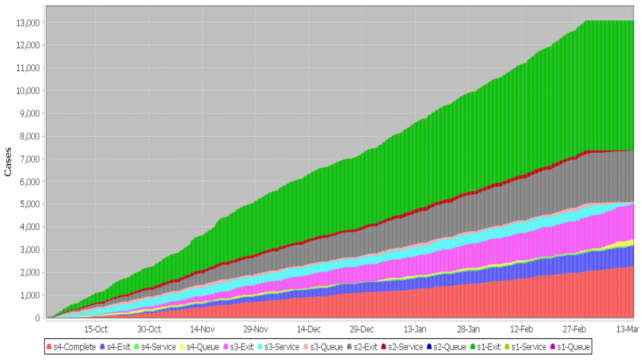


Fig. 8. CFD for the BPI Challenge 2012 log

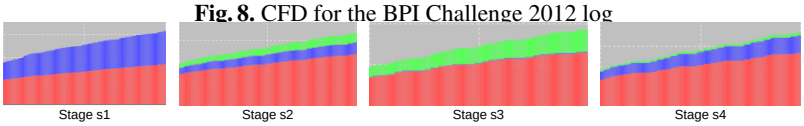


Fig. 9. Exit flows at different stages (mostly declined cases at s1, half declined and half cancelled cases at s2, mostly cancelled cases at s3 and mostly declined cases at s4)

Coming back to Q1, from the enhanced Petri net and associated performance measures, we were unable to answer Q1 as PEP does not offer any support to profile the process evolution over time. We concluded that PEP is unable to answer Q1.

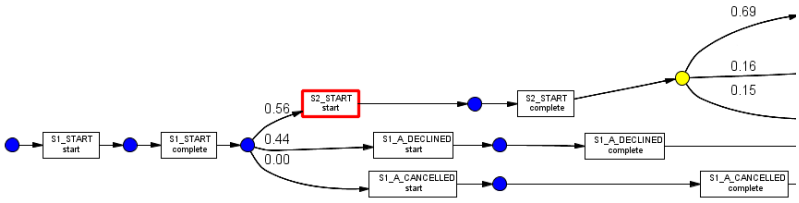
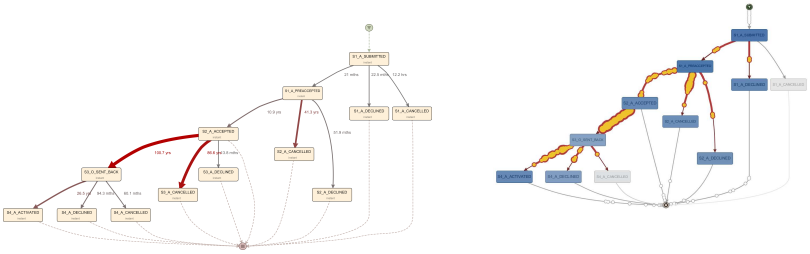


Fig. 10. Discovered Process Model in PEP, using gate events only

Disco Similar to PEP, the model discovered by Disco from the unfiltered log was rather complex, with 50 activities and over 150 paths. Hence, we also decided to retain the gate events only, leading to a rather simple model with 11 activities and 19 paths (Fig. 11.a). Based on this model, we found two ways to answer Q1. One way was using the filter by timeframe provided by Disco to select different process variants by time intervals, e.g. by months from Oct 2011 to Mar 2012. After each interval, we recorded the performance measures manually for each process variant. At the end of this procedure, we obtained a set of monthly performance measures which we could use for trend analysis. While this approach could provide a precise measurement of process evolution, the results are not easy to retrieve and interpret from the figures manually calculated. We were unable to discover any insights because of the limitation of this manual review.

Another way was to animate the log on top of the discovered model, to identify any normal and abnormal trends (Fig. 11.b). While the animation was running, we had to keep close attention to the various tokens flowing towards different directions through the model, to identify recurring patterns as well as deviations. To complete the animation for six months, it took ten minutes at maximum speed which is a reasonable time. One insight was that the cases seem to flow to the end of the process in batches.

However, it was not easy to pinpoint the recurrent timing of these batches during the animation. We were also unable to compute the volume of cases in batches due to the lack of supporting performance figures in the animation. In conclusion, we found that although the animation in Disco can provide some insightful clues with respect to process evolution, it is not possible to precisely characterize this evolution.



(a) (b)

Fig. 11. (a) The filtered process model in Disco, with highlighted bottlenecks and (b) its animation

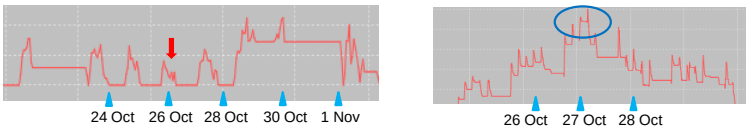
Q2: How does the formation and dissolution of bottlenecks affect the overall process performance?

SPF We can observe signs of bottlenecks on the CFD when the queue band and/or service band become wider, meaning that the process has slower response to the arrival of new cases. The formation of bottlenecks can be identified from the time series charts of CIP and TIS of the queue and service stages, particularly at the peak points. The exact measurement of these effects is provided via the on-screen tooltips and by the PST with the time interval scale. The formation of bottlenecks generally leads to an increase of CIP and TIS in the queue and service period of a stage and possibly to a decrease of FE. Conversely, these effects gradually diminish when the bottleneck dissolves.

Although the analyzed log exhibits a stable process evolution, there are signs of bottlenecks. For example, the CIP from Fig. 13.a shows that at stage s4, the queue (s4-Queue) widens from 24 Oct and at peak on 27 Oct 2011 (peak at 120 cases at 11:38am - see Fig. 12.b) and then slowly decreases onwards. The time series chart of the flow efficiency for this stage (see Fig. 12.a) also shows a fall on 26 and 27 Oct (around 0.55% fall as measured by the PST). We observe that the CIP and TIS of the s4-Service band do not increase immediately from 26-27 Oct (approximately, CIP=27 cases, and TIS=20 hours) but only afterwards (approximately, CIP=42 cases, TIS=46 hours from 29 Oct to 6 Nov 2011) as the aftermath of the previous congestion. The bottleneck slowly dissolves towards 16 Nov 2011 as the process increases its departure rate at s4-Service after the bottleneck (from ca. 20 cases/day during 23-27 Oct to ca. 24 cases per day during 28 Oct-16 Nov). The measurements show that the CIP and TIS have diminished and FE has recovered during the period 28 Oct-16 Nov (on average, CIP=36 cases, TIS=44 hours, and FE=1.35%). Similar bottleneck phenomena are evidenced in stage s4 at different times.

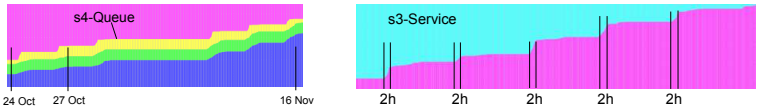
Overall, because of the two-hour working window in s3 (see Fig. 13.b), the process is mostly constrained by this stage with the most imminent queue and large CIP/TIS in the service stage.

In conclusion, with our approach it is easy to retrieve data with interpretable and precise information to answer Q2, deriving information on how bottleneck formation and dissolution affect process performance.



(a) (b)

Fig. 12. (a) Flow Efficiency at s4-Queue. (b) CIP at s4-Queue



(a) (b)

Fig. 13. (a) Example of widening queue at s4. (b) Two-hour window at s3-Service

PEP Continuing from the enhanced model in Q1, PEP can highlight the bottlenecks on the model by coloring the places of the Petri net based on their associated waiting time (see Fig. 10). This information is enriched by detailed performance measurements at the level of individual elements (see e.g. Fig.14).

However, we found no ways to reason about the impact of the formation and dissolution of bottlenecks on the process performance as the measures shown on the model are only aggregate values over the whole log timespan. It is not possible to drill down to lower levels of granularity, e.g. checking the daily arrival rate at a given place, and profile this over time. Thus, we conclude that PEP is unable to answer Q2.

Performance information of the selected place:

Frequency: 4763 visits

Arrival rate: 1.29 visits per hour

	Waiting time (hours)	Synchronization time (ho...	Sojourn time (hours)
avg	353.14	0.0	353.14
min	7.07E-3	0.0	7.07E-3
max	2149.17	0.0	2149.17

Fig. 14. Performance measures in PEP

Disco Continuing from Q1 with the discovered high-level process model, we identified two ways of detecting bottlenecks in Disco. One is displaying performance measures on the model (Fig. 11). Disco can highlight in red the exceptionally high values of activity and path durations as signs of bottleneck. We found that the paths for cancelled cases at stages s2, s3 and s4 take too long, e.g. 21 days at stage s3. In addition, the path for cases going from s3 to s4 is also longer than average (11.9 days). While through the use of filters one can measure the impact of a bottleneck on the overall process performance (e.g. by measuring how much the average cycle time improves by removing slow cases), based on the process model and the performance measures only, we did not have enough data to evaluate the extent and impact of the formation and dissolution of bottlenecks on the overall process performance.

Another way of answering Q2 is by watching the replay animation (Fig. 11). From this we can observe that there are busy flows of cancelled cases at stages s2, s3 and s4, and from s3 to s4. The tokens following these paths seem to be moving slower than those on other paths. However, we were unable to quantify these signs of bottleneck such as number of cases and waiting time, as well as the impact of these bottlenecks.

Q3: How do changes in demand and capacity affect overall process performance?

SPF The demand and capacity are represented by the arrival (AR) and departure rate (DR), respectively. The arrival rate at the first stage is the customer demand while the departure rates at different stages are their corresponding capacities. They can be observed on the CFD, as well as in the time series charts of these characteristics. Any change in these characteristics will affect the overall process performance, including the CIP, TIS and FE of the queue and service periods.

Overall, the log under exam has a much higher AR at s1-Queue as customer demand rate (79 cases/day), than DR at s4-Service as final output rate (19.23 cases/day). However, the process is able to maintain a stable evolution without congestion because there are declined and cancelled cases exiting the process at the end of a stage as demonstrated in the answer to Q1. This mechanism effectively reduces the strain of high customer demand on the process. For example, from the AR time series chart of s1-Queue (not shown for space reasons), there are a number of periods with outstandingly higher AR than the overall average AR (79 cases/day), such as 7-11 Nov, 16-18 Nov, 2-6 Jan and 27-29 Feb. However, we found that these increases have virtually no effect on later stages by reviewing the AR time series of different stages.

As such, the impact of demand and capacity is visible locally at a stage only. For example, in relation to the bottleneck reviewed in Q2, the differential chart in Fig. 15 shows that the bottleneck appears due to the stronger dominance of the arrival rate over the departure rate during that period. The difference between arrival and departure patterns has also impact on the process performance. For example, on 24 and 25 Oct, the instant AR at s4-Queue is 14 cases/day within a few hours while the instant DR is only 5 cases/day and spreads through the working day. This pattern recurs for the whole period and explains why there is a permanent long queue before s4-Service, which we identified when answering Q2. In contrast, the instant AR and DR at s2-Queue are approximately equal with the same distribution. That is why there is a very minor queue at stage s2.

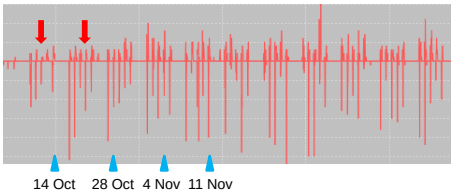


Fig. 15. Differential Chart

PEP We found no ways in PEP to investigate the impact of changes in demand and capacity on the process performance since this tool only captures one average value at every place/transition for the whole period. Hence, we are unable to answer Q3.

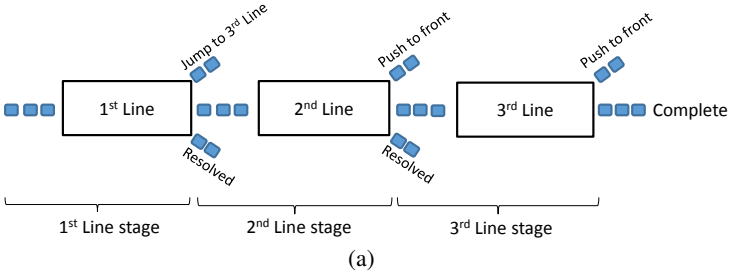
Disco We replayed the animation in Disco while focusing on the speed of the token flows at the start activity of each stage and tried to learn how this relates to the flow of tokens departing from the last activity of each stage (Fig. 11.b). However, we found it is very challenging to spot any patterns on the animation, since it is hard to capture the timing of tokens flowing at two different locations at the same time. We concluded that Disco is unable to answer Q3.

4.2 IT Incident Management Process

In this section, we evaluate our approach with the BPI Challenge 2013 log [17]. The process captured in this log is IT incident management at Volvo company. The aim of this process is to restore the IT operational activities back to normal state in the event of any incidents. Organizationally, Volvo has regional IT teams around the world divided

into three lines. The 1st Line is IT Service Desk which is to provide support for majority of standard incident handling request (IHR), such as disconnected network or disrupted system. The 1st Line receives all IHRs sent from customers. If the 1st Line is unable to solve an incident, the 2nd Line consisting of system and application specialists will be involved. In case the 2nd Line fails in dealing with the incident, the 3rd Line including experts and vendor support will be called in.

From the above context, we can take the three support lines as stages of an IHR (see Fig.16). In normal flow, an IHR sequentially passes the three lines and is finally resolved at the 3rd Line (complete case). Alternatively, it could be resolved at a previous line and then exits the process (prematurely exit). In addition, there are several exceptional flows. An IHR could be returned to a previous line, called "push to front" (PTF), if the current line considers it can be effectively resolved by a previous line. PTF is a strategy to prevent the later lines (2nd and 3rd) from being deluged with IHRs which are not their main task. In our analysis, the incidents after being pushed back to the front line will be counted as new coming IHRs. Another abnormal flow is those incidents sent directly from the 1st Line to the 3rd Line as a shortcut, labeled as "Jump to 3rd line". Some exceptional flows are not be observable in our experiment since the IHRs must follow the sequential order of the stages. However, they are only minor, such as IHRs arriving at the 3rd Line from the 1st line (2% of total cases) and IHRs coming back to the 2nd Line from the 3rd Line (1% of total cases). By taking the three lines as stages, we are analyzing the process flow from an organizational perspective with a focus on the performance of support lines which is also a question concerned by Volvo at the BPI Challenge 2013.



(a)

Fig. 16. IHR Flow

Below we report the experiment with our approach only. For PEP and Disco, the techniques of using them to answer the three questions are the same as described in our previous experiment which result in the same observations.

From Sun, 15 Apr 2012 04:49:40 +0200 To Sat, 5 May 2012 20:23:56 +0200 (20 days) (Mean/Median)

	AR(cases/day)	DR(cases/day)	ER(cases/day)	CIP(cases)	TIS(hours)	FE(%)
System	345.77 / 72.00	1.84 / 0.00	296.61 / 0.00	1186.74 / 1080.50	528.37 / 533.50	14.54
1st-queue	345.77 / 72.00	345.77 / 72.00		14.41 / 3.00	2.13 / 0.00	
1st-service	345.77 / 72.00	307.55 / 48.00	211.84 / 0.00	755.57 / 694.50	112.31 / 133.00	9.28 / 9.45
2nd-queue	95.71 / 24.00	92.71 / 24.00		81.81 / 67.00	30.26 / 24.00	
2nd-service	92.71 / 24.00	86.90 / 0.00	83.61 / 0.00	310.00 / 290.00	118.81 / 129.00	19.71 / 19.66
3rd-queue	3.29 / 0.00	2.90 / 0.00		3.77 / 3.00	42.52 / 35.50	
3rd-service	2.90 / 0.00	1.84 / 0.00	1.16 / 0.00	21.18 / 23.00	222.34 / 212.00	14.63 / 14.96

Fig. 17. Performance Summary of BPIC2013 Log

Q1: How does the overall process performance evolve over time?

The CFD in Fig.18 shows a remarkable transition from slow to sharp evolution at the 1st and 2nd Line. Exact measurements can be taken from the summary table (Fig. 17) for different time intervals. The slow evolution takes place from the starting time of the log (31 Mar 2010) to 15 April 2012 (average AR-1st line and DR-1st line are approximately 1 case/day, average AR-2nd line and DR-2nd line are approximately 0.5 case/day) and the sharp evolution period is from 15 April 2012 to 5 May 2012 (average AR-1st line is 345 case/day, DR-1st line is 307 case/day, average AR-2nd line is 96 case/day, DR-2nd line is 87 case/day). Particularly, the arrival and departure rate at the 1st and 2nd Line are extremely high from 2 to 5 May 2012. Conversely, the 3rd Line has maintained slow evolution throughout the period of analysis (average AR and DR are comparable at 3 cases/day during the busiest period from 23 April to 15 May 2012).

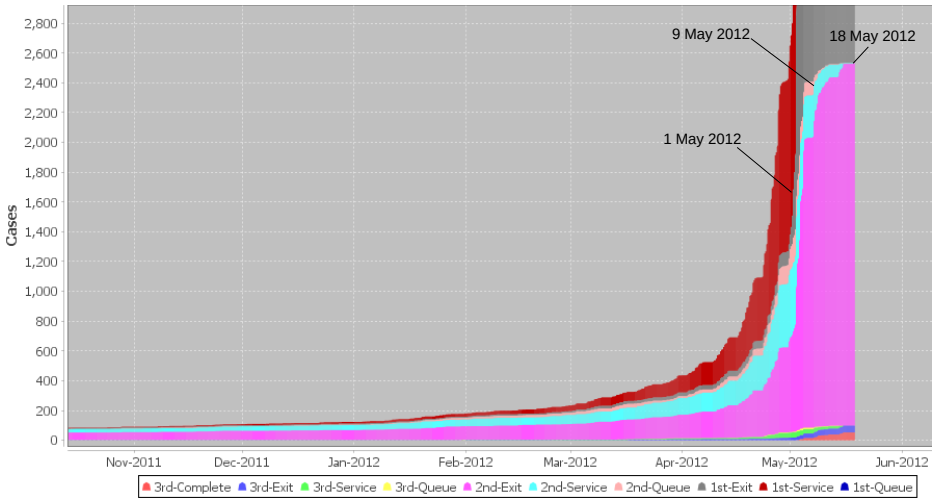


Fig. 18. CFD for the BPI Challenge 2013 Incident Log

Both the 1st and 2nd Line had considerable exit flow during the sharp evolution. During this period, the departure flow of the 1st Line consisted of 40% resolved incidents, 35% "jump to 3rd line" and 25% transfered to the 2nd Line. At the same time, the 2nd Line had a departure flow with 67% "push to front", 31% resolved and 2% transfered to the 3rd Line. Note that the 2nd Line has maintained a "push to front" behavior from the beginning. Only until 2 May 2012 where there was a sharp increase of demand at the 1st Line (see Fig. 19), it started to have incidents resolved. This indicates that the 2nd Line has insisted in pushing incidents back to the 1st Line and was only involved in handling incidents when the 1st Line faced an exceptionally high pressure on 2 May 2012. Even though, a larger proportion of cases was sent back from the 2nd to the 1st Line after 2 May 2012.

In addition, by observing the departure flow at every line, we see that at the 1st Line the exiting cases until 1 May 2012 were either furthered to the 2nd Line or sent to the 3rd Line and remarkably there were no resolved cases before 1 May 2012. There were also few resolved cases at the 2nd Line before 1 May 2012 (totally 13 resolved incidents). Similarly, cases resolved at the 3rd Line were only from 1 May 2012. It seems that the 1st Line had accumulated a large number of unresolved incidents until 30 April 2012 and then had to release them from 1 to 18 May 2012 by resolving 40% and transferring the rest to the 2nd and 3rd Line. From 1 to 18 May 2012, although the

2nd and 3rd Line were more active than before in resolving incidents, they continued returning many cases back to the front line. Coming the end of the sharp evolution, from 9 to 18 May, the pressure was significantly reduced at support lines, with average AR of approximately 8 case/day at the 1st Line, 4 case/day at the 2nd Line, and 0.6 case/day at the 3rd Line.

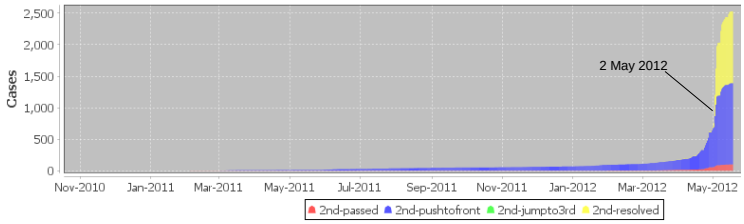


Fig. 19. Departure Flow at the 2nd Line (BPIC2013 Incident Log)

Q2: How does the formation and dissolution of bottlenecks affect the overall process performance?

There are several times of congestion, such as the queue at the 2nd Line on 28 April 2012 (see Fig. 20).

The queues at the 2nd Line were usually formed by the high arrival rate on Friday while there was little work at the weekend; thus, the queue of pending cases became widening. This also increased the TIS of the whole process (the increase is 69 hours on 27 April 2012 and 37 hours on 1 May 2012). However, we observe that the 2nd Line has also improved its processing rate after those bottlenecks. In particular, on 30 April 2012, after the weekend, the queue persisted because AR still exceeded DR. It only disappeared when the 2nd Line increased its departure rate to match with the AR. As a result, the queue gradually disappeared and TIS has diminished substantially after 2 May 2012.

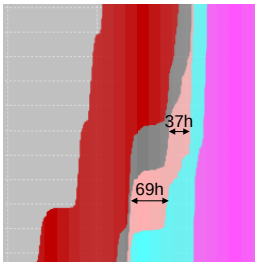


Fig. 20. Bottleneck at the 2nd Line (BPIC2013 Incident Log)

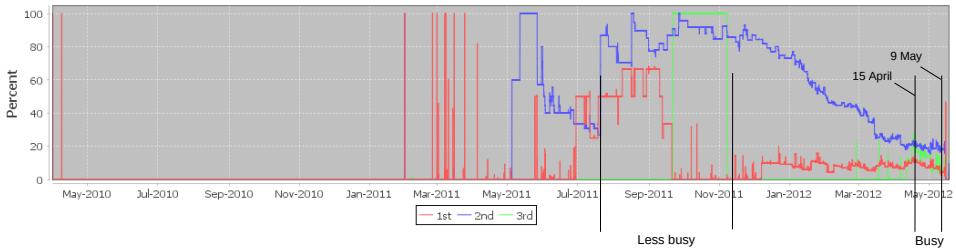


Fig. 21. Flow Efficiency Chart (BPIC2013 Incident Log)

In terms of flow efficiency, it can be recognized from Fig. 21 that during the period of sharp evolution, the flow efficiency of all support lines was at low level as opposed to

the very high level when the teams were much less busy (as arrival rate was very low). This is because if the service desk teams have to deal with many incident handling requests (IHR) simultaneously, each IHR will be treated at slower pace and has more idle time. It also reflects that the flow efficiency is often very hard to achieve in parallel with the resource utilization efficiency [3], meaning if resources are highly utilized, the flow efficiency will usually suffer.

Q3: How do changes in demand and capacity affect overall process performance?

The busiest period of the process was from 15 April to 6 May 2012. The process encountered a surge of demand in the second half of April 2012 and extended to early May, particularly at the 1st and 2nd Line.

In response, the 1st Line acted rapidly during the period of 2-3 May as shown with very quick departure flow. The DR-AR differential chart of the 1st Line service stage (Fig.22) shows that the capacity did not meet the demand rate until 1 May 2012. As a consequence, the 1st Line during this period was characterized by large CIP (21 cases on average) and TIS (1,566 hours on average). After 1 May 2012, the capacity increased and exceeded the demand rate. Thus, during this period, the 1st Line had large CIP (299 cases on average) and small TIS (80 hours on average). We also observe that the 2nd Line has similar behavior to the 1st Line during this period.

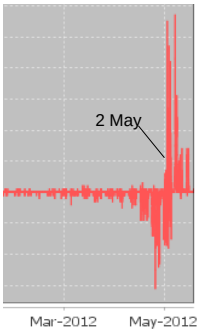


Fig. 22. Differential Chart at the 1st Line (BPIC2013 Incident Log)

In general, it can be observed that the 1st and 2nd Line exhibit a matching capacity in response to the sharp fluctuation of the demand. Particularly, the departure rate has been boosted to respond to the increasing work load when the demand suddenly increased.

5 Conclusion

We presented an approach to analyze flow performance from event logs based on the concept of SPF and associated characteristics and visualizations, which transpose ideas found in lean management and agile software development to the field of PPM. The evaluation on real-life event logs puts into evidence qualitative advantages of this approach with respect to existing PPM techniques.

The proposed approach is not without limitations. One limitation is the assumption that the log is divided into stages. A direction for future work is to design techniques for automated identification of *candidate stages* from a log for flow performance analysis.

Another limitation is that the approach still requires the user to manually identify patterns from the stage characteristics and visualizations, particularly patterns associated with formation and dissolution of bottlenecks. There is an opportunity to extend the SPF approach with techniques from statistical process control and change point analysis, such as CUSUM charts [20], to support the identification of such patterns.

Another future work avenue is to conduct empirical evaluations of the SPF approach for example via controlled experiments, in order to assess its relative advantages with respect to existing PPM approaches and to validate major design choices, such as the choice of stage characteristics and visualizations.

Acknowledgments. NICTA is funded by the Australian Government via the Department of Communications. This research is funded by the Australian Research Council Discovery Project DP150103356 and the Estonian Research Council.

References

1. van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer (2013)
3. Modig, N., Ahlström, P.: This is lean: Resolving the efficiency paradox. Rheologica (2012)
4. Hornix, P.T.: Performance analysis of business processes through process mining. Master's thesis, Eindhoven University of Technology (2007)
5. Gunther, C.W., Rozinat, A.: Disco: Discover your processes. In: Proc. of BPM Demos. Volume 940 of CEUR Workshop Proceedings. (2012) 40–44
6. van der Aalst, W., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**(2) (2012) 182–192
7. van Dongen, B.F., Adriansyah, A.: Process mining: fuzzy clustering and performance visualization. In: Proc. of BPM Workshops, Springer (2010) 158–169
8. Rebuge, A., Ferreira, D.: Business process analysis in healthcare environments: A methodology based on process mining. Information Systems **37**(2) (2012) 99–116
9. de Leoni, M., van der Aalst, W.P., Dees, M.: A general framework for correlating business process characteristics. In: Proc. of BPM, Springer (2014) 250–266
10. Pika, A., Wynn, M.T., Fidge, C.J., ter Hofstede, A.H., Leyer, M., van der Aalst, W.M.: An extensible framework for analysing resource behaviour using event logs. In: Proc. of CAiSE, Springer (2014) 564–579
11. Gunther, C.W., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: Proc. of BPM, Springer (2007) 328–343
12. Conforti, R., Dumas, M., Rosa, M.L., Maaradji, A., Nguyen, H., Ostovar, A., Raboczi, S.: Analysis of business process variants in apromore. In: Proceedings of the BPM Demos. Volume 1418., CEUR (2015)
13. Senderovich, A., Weidlich, M., Gal, A., Mandelbaum, A.: Queue mining - predicting delays in service processes. In: Proc. of CAiSE, Springer (2014) 42–57
14. de Smet, L.: Queue mining: Combining process mining and queueing analysis to understand bottlenecks, to predict delays, and to suggest process improvements. Master's thesis, Eindhoven University of Technology (2014)
15. van Dongen, B.F.: BPI Challenge 2012. Eindhoven University of Technology. Dataset (2012) http://dx.doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.
16. Reinertsen, D.: Managing the Design Factory: A Product Developers Tool Kit. Simon & Schuster Ltd. (1998)
17. Ward, S.: BPI Challenge 2013. Ghent University. Dataset (2013) http://dx.doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07.
18. Nguyen, H., Dumas, M., ter Hofstede, A., La Rosa, M., Maggi, F.: Business process performance mining with staged process flows. Qut eprints technical report 91110 (`http://eprints.qut.edu.au/91110`), Queensland University of Technology (2015)
19. Venkatesh, V., Bala, H.: Technology acceptance model 3 and a research agenda on interventions. Decision Sciences **39**(2) (2008)
20. Reynolds, M., Amin, R., Arnold, J.: CUSUM charts with variable sampling intervals. Technometrics **32**(4) (1990) 371–384