# Multi-Dimensional Performance Analysis and Monitoring Using Integrated Performance Spectra

Vadim Denisov
Eindhoven University of Technology
The Netherlands
Email: v.denisov@tue.nl

Dirk Fahland
Eindhoven University of Technology
The Netherlands
Email: d.fahland@tue.nl

Wil M.P. van der Aalst
Department of Computer Science,
RWTH Aachen, Germany,
Email: wvdaalst@pads.rwth-aachen.de



***Abstract*—In process mining, basic descriptive statistics over observed events of event logs or streams, projected onto a process model, are typically used for performance analysis. The so-called performance spectrum is used for the fine-grained description of process performance over time, additionally revealing phenomena related to the behavior of multiple cases together. The performance spectrum computed from traces aligned with a process model allows performance analysis of processes with concurrency. However, performance spectra are used to describe performance only along the case level, leaving performance analysis and monitoring of other process dimensions out of scope. This paper presents an approach and tool combining a synchronous proclet system with a performance spectrum for multi-dimensional performance real-time monitoring and post-mortem analysis. While the tool is a proof-of-concept implementation, designed for analysis of the control-flow, resource and queue dimensions of logistic processes, the presented concepts are general.**

## I. Introduction

Performance analysis is an important element in process management relying on precise knowledge about actual process behavior and performance for enabling improvements [1], and for detecting and mitigating performance deviations in the real-time settings [2]. Existing techniques project *aggregated* performance diagnostics, based on descriptive statistics over observed events of event logs or streams, onto a process model visualization [3]. The common problem of those techniques is the inability to reveal phenomena related to the behavior of multiple cases together, such as *overtaking* (when a case bypasses cases that started earlier), *batching*, *queueing* and so on [4]. As a result, causations and correlations between instances of those phenomena cannot be revealed and analyzed, e.g., a longer than usual average duration of a particular step of a process cannot be explained by averages or assuming independence of cases.

To address these limitations, the so-called *Performance Spectrum* (PS) was introduced in our prior work [2], [5], and implemented in the Performance Spectrum Miner (PSM) [6]. A tool integrating the PS with a Petri net model, capable of handling *concurrent* processes, is presented in [4]. However, both techniques [4], [6] visualize performance only along the case level. While this helps identifying process steps and moments in time with performance problems, the analyst cannot investigate the environment that causes performance problems. For example, a Material Handling System (MHS), such as an airport Baggage Handling System (BHS), can be considered as a *network of queues and resources* connected according to the system physical layout [7]. The behavior of a case, e.g., a bag (control-flow), queues, and resources lie in different process dimensions that are usually studied separately. As a result, such a separate analysis does not reveal how, for example, resource availability affects queue waiting times, and so on. To study their interplay, the analyst has to understand how the control-flow, queue, and resource dimensions are interconnected. For that, an *integrated analysis* of *all* the dimensions is required (**Req1**). In turn, this requires a *process model* that describes the behavior of all the dimensions together, so that the performance visualization along all dimensions over time is presented in an integrated context. Industrial adoption requires simple models that describe process dynamics using domain concepts. At the same time, process analysts may require more precise formal models, so a model should be visualized in two forms: (1) as a *domain-specific visualization* for domain experts, and (2) as a *formal model* for process analysts (**Req2**). Further, understanding process performance problems has two use cases: (1) post-mortem analysis of offline data, e.g., for root-cause analysis or for developing predictive models, (2) online visualization to let operators observe performance problems at run-time in a detailed manner. While the former is addressed in [2], [4]–[6], [8], the latter is not, so visualization of the performance description over time in *real-time* is required (**Req3**).

We present a tool[1] that allows an integrated analysis and visualization of performance problems along the dimensions of control-flow, queues, and resources for the domain of MHSs. It is a *proof-of-concept* implementation, which can use a real-time event stream from the provided BHS simulation model or a real MHS. We address the problem of *multi-dimensional performance analysis* using a synchronous proclet system [9] called a PQR-system [7] that describes the *Process*- (P-), *Queue* (Q-) and *Resource* (R-) dimensions of an MHS. The tool allows to study Performance Spectra (PSa) of those dimensions in the PSM, using a GUI of the PQR-system visualization for selecting a required PS fragment of the required dimension to be shown. We mitigate the problem of understanding a process model by aligning its layout with Material Flow Diagrams (MFDs), the

[1] The source code, video and further documentation available on https://github.com/processmining-in-logistics/psm/tree/pqr

visualization widely used in the MHS domain, and by hiding implementation details when needed. We support real-time monitoring of the system performance by (1) token animation over the domain specific system visualization, similar to token animation in most process mining tools, and (2) through the PS being updated in real-time to show how the PS-based monitoring allows faster and more accurate analysis. Finally, to illustrate our concept in action, we show how a multi-dimensional analysis with PSa allows identifying root causes of performance outliers.

The remainder is organized as follows. Section II introduces PQR-systems and PSa. Section III explains how the tool satisfies the requirements of this section, and Section IV discusses evaluation and concludes the paper.

## II. PQR-Systems and Performance Spectra

In MHSs, materials (e.g., bags in BHSs) are typically moved by *conveyors*. In Fig. 1(a) the conveyor moves bags from location $a$ to location $b$, the system tracks bags at those locations through sensors that detect the front and back sides of each bag. We model MHSs as a synchronous proclet system called a PQR-system, where the only P-proclet models the conveyor layout, Q-proclets model conveyors as queues, and R-proclets model resources. The life-cycle transitions *start* and *complete* model how the front and back sides of a bag pass a location. A PQR-system example is shown in Fig. 1(b). The P-proclet (shown in red) shows the logical layout of the system of Fig. 1(a). The Q-proclet $a : b$ (shown in blue) models bag transportation as a FIFO queue with the minimal waiting time $t_{\mathrm{wQ}}$. The R-proclets $a$, $b$ (shown in green) model resources handling bags at the conveyor beginning (location $a$) and end (location $b$) as a single server (R-proclet of the resource $b$ is not shown to save space). Each resource has the minimal service time $t_{\mathrm{sR}}$ required for handling a case (bag), and the minimal waiting time $t_{\mathrm{wR}}$ for becoming ready for handling the next case. The *channels* shown by dashed lines define how transition occurrences of different proclets synchronize. As a result, each event is a part of three traces: of a case, of a queue, and of a resource participating in the same step.

The PS is computed for *segments*. A segment is a pair of activities $(x, y)$ where $y$ directly follows $x$. It describes a *step* from activity $x$ to activity $y$, e.g., a bag moving from location $x$ to $y$. Each case advancing from activity $x$ to $y$ generates an *occurrence* of the segment $(x, y)$, which is characterized by two timestamps $(t_x, t_y)$. In Fig. 1(c), the PS of the PQR-system of Fig. 1(b) is shown. Its top segment $(a_s, b_s)$ contains two segment occurrences of cases $pid_1$ and $pid_2$ with timestamps $(t_1, t_2)$ and $(t_3, t_4)$ respectively. Multiple segments can be composed into the *detailed PS* [5]. Fig. 1(c) additionally shows the PS for segments of the queue $a : b$ and resource $a$. During an occurrence of $(a, b)$ in the P-proclet for case $pid_1$, the resource $a$ handles it (occurrence $o_1^1$), then waits (occurrence $o_1^2$) for the next case. As soon as the case is handled by $a$, it is enqueued into the queue $a : b$, where it waits (occurrence $o_1^5$) for handling by the resource $b$. Case $pid2$ is handled similarly (occurrences $o_2^3$, $o_2^4$ and $o_2^6$ correspondingly).

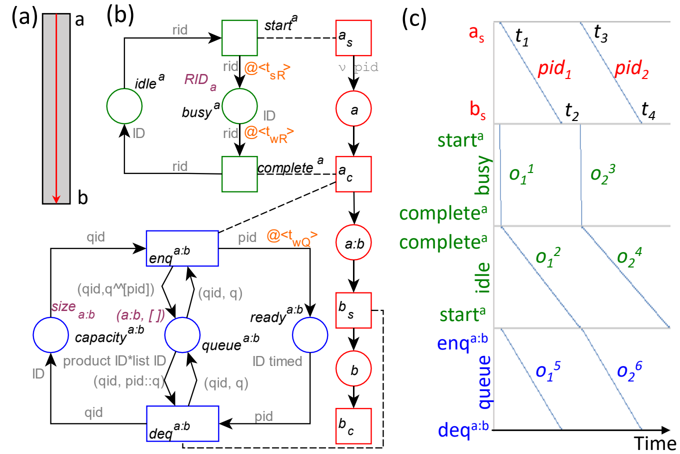

Fig. 1. The MHS conveyor (a), corresponding PQR-system (b) and PS (c).

## III. Multi-Dimensional Analysis and Monitoring

This section is organized as follows. We first introduce the main components of the tool. Then we explain how to interpret performance spectra of P-, Q- and R-dimensions of a PQR-system, illustrate how we adapt the visualization of the process model to needs of domain experts, and consider pros and cons of performance analysis and monitoring using classical token animation versus visualization of PSa. Finally, we formulate a brief guideline for performance analysis using an environment with the integrated PS and process model that implements the requirements of Section I.

### A. Tool Components

Our tool consists of the *PQR-system*, the *PSM*, and the optional *BHS simulation model* of a BHS (Fig. 2). The PQR-system *visualizes* the process model and provides a GUI for filtering segments of different dimensions visualized by the PSM. The simulation model provides a GUI to control simulation scenarios and their animation. It sends each segment occurrence to the PSM as a *datagram* to simulate real-time monitoring of remote systems. In practice, the data for the tool can also be streamed from a real system.

### B. Performance Spectra of Multiple Dimensions

As we discussed in Section I, PQR-systems describe P-, Q- and R-dimensions. Events, generated by the simulation model, have attributes that carry identifiers for corresponding case notions and transition labels, so the PSM can assemble traces per case (bag), queue, and resource for computing and visualizing their PSa, while the PQR-system GUI allows to choose segments to be shown.

Fig. 1(b) shows the PS corresponding to "normal" work, i.e., without performance outliers. In contrast, Fig. 3 shows a "slow" bag (occurrence $o_1$), surrounded by "faster" bags. In the model, we see that either resource $a$ or $b$, or queue $a : b$ could cause the delay. However, considering the PS of just the control-flow dimension is insufficient to give a concrete answer. For that, we also analyse the PSa of the Q- and R-dimensions.

We start from the PS of resource $a$. Each resource of a PQR-system defines exactly one trace of its start and complete

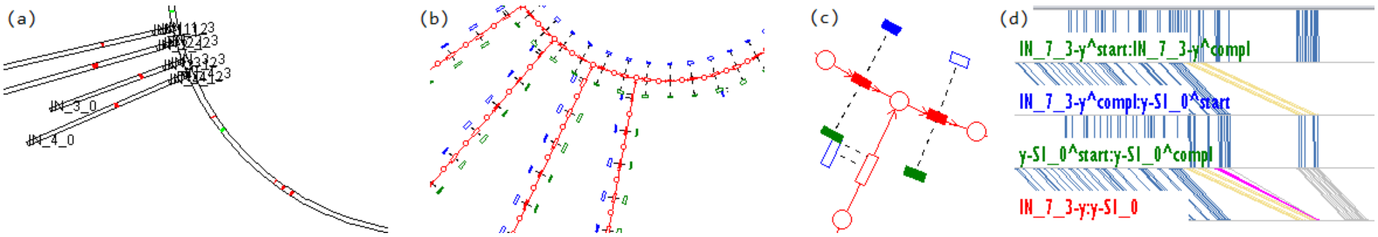

Fig. 2. The BHS simulation model animation frame (a), the PQR-system visualization (b,c), and the PS visualization (d).

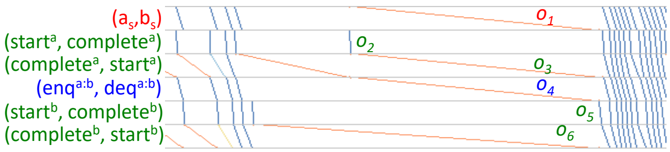

Fig. 3. The PS of the P-, Q- and R-dimensions.

activity events. Thus, the segment $(start^a, complete^a)$ represents the serving (handling) of one bag (state busy), while the segment $(complete^a, start^a)$ represents the absence of bags to handle (state idle). In the PS, we see that the service time of the delayed case (occurrence $o_2$) is similar to the service time of the surrounding cases, while the following longer idle time (occurrence $o_3$) indicates absence of cases to be handled. We conclude this resource did not cause the delay but had to wait longer for the next case. Similarly, resource $b$ had to wait longer (occurrence $o_6$) to start handling the delayed case (occurrence $o_5$), but the handling itself was not delayed. Further, we analyse the PS of queue $a : b$. Similarly to the resources, each queue defines exactly one trace. A segment of a queue represents an element (bag) waiting in the queue. In the PS of queue $a : b$, we see the longer waiting time for the delayed bag (occurrence $o_4$), so the queue indeed caused this delay. That proves also a longer waiting time of $b$ ($o_6$). Further in Section III-E we will show how to identify which system queue or resource *initially* caused a performance outlier.

### C. Adaptive Visualisation of PQR-systems

In Section I, we highlighted challenges of the use of process models in process mining tools oriented on diverse users. To overcome them, we followed two directions: (1) making models similar to process representations the domain experts are used to, i.e., MFDs, and (2) hiding information that is irrelevant for the current analysis phase or already known to the analyst.

For that, we aligned the PQR-system layout with the layout of the system MFD. In our tool, an MFD is present by the simulation model visualization (Fig.2(a)). We repeat this layout in our P-proclet, thereby making it easily recognizable by everybody familiar with the MFD. Then, we assume that implementation details of the Q- and R-proclets are extremely useful for getting to know the PQR-system first, but verbose afterwards. So we provide options to hide various details and dimensions.

### D. Token Animation versus Performance Spectra

As we discussed in Section I, token animation over a system model is widely used for performance analysis and monitoring. Our simulation model also uses it to show a current location of each bag in the system. Additionally, it highlights stopped (blocked) conveyors in red. The PSM draws and classifies segment occurrences as soon as the corresponding events are generated by the system, and automatically scrolls the view towards the latest segment occurrence. As for ongoing segments the end activities and timestamps are not observed yet, it estimates them only if there is no choice in the control flow, i.e., when the end activity is known, using the observed minimal duration of process segments.

Let us first consider two screenshots of animation in the time moments $t_1$ and $t_2, t_1 < t_2$ in Fig. 4(a,b). The first screenshot shows normal work (no stopped (red) conveyors). In the second screenshot, the bags are not moving, because the conveyors are stopped (red). While the states are clear, a painstaking analysis of dozens or even hundreds of frames in between is required to understand *how* and *why* the incident was developing between those time moments. For processes with many cases executing simultaneously, and with a wide range of average durations of different process steps, such an approach is hardly feasible. In contrast, the PS allows to see the performance dynamics of the system for a time interval of interest *at once*, speeding up the detection of outliers during performance analysis or monitoring. In Fig. 4(c) we instantly see the delay in bag handling in segments $s1, s3-s9$, and how it propagates through the segments over time. As soon as an outlier is detected, the PS analysis can reveal its root causes, as we show next.

### E. Using Integrated Performance Spectra and PQR-systems

In the tool, the PQR-system and PSM are integrated into a single GUI. The PQR-system panel allows adding and removing segments of the P-, Q- and R-dimensions to the PS in the PSM, and the PSM allows their ordering needed for a particular analysis. Additionally, the PSM allows navigation from a PS segment back to the corresponding place of the PQR-system. For example, in Fig. 4(c) the PSM shows (1) process segments $s2 - s9$ forming a route from the BHS check-in link $IN_4$ toward unit $y - S_0$ diverting bags to scanner $S1$, (2) resource segments $s10$ and $s11$ representing the resource of merge unit $IN_7^3$ preceding directly diverting unit $y - S_0$, and (3) queue segment $s12$ representing the queue between those merge and diverting units. Here we provide a basic guideline for the analysis of performance deviations, focused on *blocking* (a delay in performing of a particular process step) and *high load* (a higher number of cases for a process segment within a given time window). The guideline consists of the following steps:

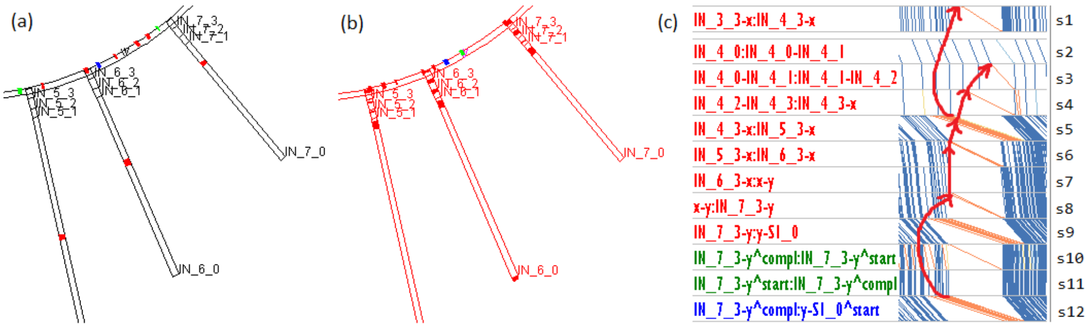


Fig. 4. The token animation frames showing normal work (a) and blocking (b), and the blocking propagating in the PS (c).

1. *Spotting a deviation.* For that, we explore the P-proclet PS visually, e.g., in Fig. 4(c) we see longer occurrences (blockings) in segments $s1, s3 - s9$, shown in orange. We select one of them, segment $s3$, for the further analysis.
2. *Identifying the starting segment.* For that, we analyze the PS, starting from the selected segment, *against* the direction of the outlier propagation. Concretely, high load typically propagates *forward* along the control flow, and blocking propagates *backward*. We add each next segment, containing the same type of deviation around the time of the previously spotted one, to the chain of segments until the next segment does not have deviations. The last segment of this chain had caused the whole chain of outliers. In our example, we have the chain $\langle s3, \ldots, s9 \rangle$. $s9$ is the last segment where the blocking is still observed, so it caused the whole chain of blocking.
3. *Analysing the other dimensions.* For the starting segment, we analyze the PS of the *Q*- and *R-dimensions*. For blocking, a longer queue waiting time or a longer resource service time can cause the delay, while for high load a higher number of cases accumulated, for example, during a recent blocking can cause a load peak. In our example, we study the R- and Q-dimensions for segment $s9$. Segment $s11$ (state busy) has a long pause, i.e., it did not handle cases during the deviation time. Segment $s10$ (state idle) has a single long occurrence, i.e., a longer waiting time before the next case. Segment $s12$ (queue) shows multiple delayed traces whose waiting times are longer than the waiting time of the surrounding traces, so we conclude this queue caused the delay in $s9$ and the whole chain of blocking as well.
4. *Domain-specific explanation*. Using domain knowledge, the analysis results can be explain in terms of the underlying process or system. In our example, conveyor $IN_7^3 - y : y - SI_0$ could be blocked by a bag, or it could have a technical malfunction, causing longer waiting times for all cases (bags) located on it. In turn, the preceding conveyors were gradually stopped one by one as they could not hand over bags to the further (already stopped) conveyors. The red arrows in Fig. 4(c) shows the incident development. Note, after $s5$ blocking propagates via merge unit $IN_4^3 - x$ in *two* directions: to $s4$ and $s1$.

## IV. Conclusion

The tool combining PSa and MFDs was successfully evaluated at Vanderlande for several systems and use cases, still in the off-line mode. We showed that such an integration allows a shallow learning curve and faster performance analysis for domain experts. We expect further evaluations of the combined PQR-systems and PSa for both analysis and monitoring. Our tool still lacks automatic detection of performance patterns [10], projecting descriptive statistics onto the model [3] and a more robust approach for estimation of end activities and timestamps of ongoing segments. The latter is the subject of future work.


## Acknowledgment

The research leading to these results has received funding from Vanderlande in the project “Process Mining in Logistics”.


## References


[1] L. Maruster and N. R. T. P. van Beest, “Redesigning business processes: a methodology based on simulation and process mining techniques,” *Knowl. Inf. Syst.*, vol. 21, no. 3, pp. 267–297, 2009.

[2] V. Denisov, D. Fahland, and W. M. P. van der Aalst, “Predictive performance monitoring of material handling systems using the performance spectrum,” in *ICPM 2019*. IEEE, 2019, pp. 137–144.

[3] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[4] W. M. P. van der Aalst, D. Tacke Genannt Unterberg, V. Denisov, and D. Fahland, “Visualizing token flows using interactive performance spectra,” in *Petri Nets 2020*. Cham: Springer International Publishing, 2020, pp. 369–380.

[5] V. Denisov, D. Fahland, and W. M. P. van der Aalst, “Unbiased, fine-grained description of processes performance from event data,” in *BPM 2018*, ser. LNCS, vol. 11080. Springer, 2018, pp. 139–157.

[6] V. Denisov, E. Belkina, D. Fahland, and W. M. P. van der Aalst, “The performance spectrum miner: Visual analytics for fine-grained performance analysis of processes,” ser. CEUR Workshop Proceedings, vol. 2196. CEUR-WS.org, 2018, pp. 96–100.

[7] V. Denisov, D. Fahland, and W. M. P. van der Aalst, “Repairing event logs with missing events to support performance analysis of systems with shared resources,” in *Petri Nets 2020*. Cham: Springer International Publishing, 2020, pp. 239–259.

[8] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés, “Predictive monitoring of business processes: A survey,” *IEEE Transactions on Services Computing*, vol. 11, no. 6, pp. 962–977, Nov 2018.

[9] D. Fahland, “Describing behavior of processes with many-to-many interactions,” in *Petri Nets 2019*, ser. LNCS, vol. 11522. Springer, 2019, pp. 3–24.

[10] E. L. Klijn and D. Fahland, “Performance mining for batch processing using the performance spectrum,” in *BPM Workshop 2019*, ser. LNBIP, vol. 362. Springer, 2019, pp. 172–185.