BPIC’2018: Mining Concept Drift in Performance Spectra of Processes

Citation for published version (APA):

DOI:
10.4121/uuid:3301445f-95e8-4ff0-981f1f204972

Document status and date:
Published: 01/01/2018

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
BPIC’2018: Mining Concept Drift in Performance Spectra of Processes

Vadim Denisov¹, Elena Belkina, and Dirk Fahland¹

Eindhoven University of Technology, The Netherlands, v.denisov@tue.nl, e.belkina@hotmail.com, d.fahland@tue.nl

Abstract. The BPI Challenge 2018 is focused on analysis of a process that covers the handling of applications for EU direct payments for German farmers from the European Agricultural Guarantee Fund. This work is focused on performance analysis of this process. Our goal is to demonstrate and explain a novel process mining technique and tool, which we recently created, by answering questions of this challenge. Thus we target the Academic Category, demonstrating applicability of the technique for descriptive performance analysis, rather than answering the questions of the challenge in more detail.

Keywords: process mining, performance analysis, performance spectrum, visual analytics, performance patterns

1 Introduction

The BPI Challenge 2018 poses the task of answering several business questions about a process for handling of applications for EU direct payments for German farmers from the European Agricultural Guarantee Fund through analyzing the event data of all cases processed in the years 2015-2017.¹

In this report for the “academic category”, we focus on analyzing business question 4: “Usually, around the same number of applications from the same farmers is handled every year. The processes should be similar each year, but may differ due to changes in regulations or in their technical implementation. How can one characterize these differences as a particular instantiation of concept drift?”² Concept drift describes situations where processes are not in steady state but processing of cases changes temporarily, seasonally, or permanently. Changes may be so significant that a single model no longer adequately describes the process, but rather multiple models are required to describe the different populations.³ Existing works on concept drift focus on detecting changes in the control-flow of cases, i.e., the steps and routing of a single case for descriptive [3,5] and predictive [6] analysis. Changes in process performance cannot be detected with model-based process mining-based performance analysis [2] as these aggregate the waiting and processing times of events of all cases per activity or segment between two activities.

¹https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972
In the following, we set out to show that a recent visualization of event data called the performance spectrum [4] allows to describe further types of concept drift in the control-flow and in the performance perspective. The performance spectrum, described in more detail in Sect. 2, explicitly visualizes the performance of each process step in each case over time as a line. The angle of the line describes the performance of the step in this case. The relation of angles and density of multiple lines of different steps over time gives rise to various patterns describing changes in performance and in synchronization behavior of different cases at a very fine granular level. Through the visualization and a taxonomy for the patterns (see Appendix A and [4]) we will show that the process faces

- concept drift in the handling of all cases together such as changes of prioritization of cases, or changes in batching of cases at various parts of the process over the different years, and
- concept drift in the performance perspective of the process describing significant changes both in individual steps as well as in the overall process performance in terms of performance patterns [4].

With the change of replacing parcel documents by geo-parcel documents in 2016, we found that the processing became more homogeneous and cases advance at a more steady pace, most notably through new batching behavior prior to the creation of the (geo-)parcel documents, and a slower and more steady processing after the creation of the (geo-)parcel documents. Although processing is in some more parts slower, the share of too late payments does not increase over the years. The slower and more steady processing however has the positive effect that less activities have to be cancelled and repeated, most notably the share of decision to be revised dropped to 2% (from 38%) and the share of payments to be cancelled and initiated again dropped to 13% (from 100%). A more extensive summary of our findings is presented in Sect. 6.

In the remainder, we first briefly describe the performance spectrum and its implementation in Sect. 2. We give an overview on the process through the provided event data in Sect. 3 and through the performance spectrum in Sect. 4 to plan our analysis. Sect. 3 describes our pre-processing of the event data which we use in Sect. 4 to analyze the concept drift of the process. We conclude in Sect. 6.

2 Our Technique and Tool

In this section we explain the concept of the performance spectrum by example and review main features of the tool we have developed for work with performance spectra of processes.

Recently introduced performance spectrum [4] maps all observed flows between two process steps together regarding their performance over time. The tool for working with performance spectra of processes has been developed as an interactive ProM plugin the Performance Spectrum Miner (PSM) in package “Performance Spectrum” with source code and further documentation available at https://github.com/processmining-in-logistics/psm.
The main window of the PSM is shown in Fig. 1. It consists of two parts: the scrollable main panel (1) and the control panel (2). During an analysis session in the PSM, a user first imports and pre-processes an event log, providing pre-processing parameters, which are explained further in this section, then analyzes an obtained performance spectrum in the main panel. A performance spectrum consists of segments, that represent observed flows between two process steps over the time axis. It can be

Fig. 2. Examples of a detailed (a) and aggregated (b) performance spectrum.
A detailed performance spectrum shows information about individual traces. For instance, in Fig. 2a) segment Z2 represents a step between activities Create Fine and Send Fine, and has name Create Fine:Send Fine. Each spectrum line within the segment, e.g. highlighted line AB, represents occurrences of Create Fine that are followed by Send Fine. Occurrences of activities in points A and B have timestamps Ta and Tb correspondingly. Similarly, within Z3, line BC represents a case that has activity Send Fine, which is directly followed by activity Insert Fine Notification, which has timestamp Tc. Angles of lines indicate duration of steps: vertical lines show instant execution, while sloping lines indicate slower execution. The colors of lines show performance classes, assigned by a selected classifier. Available classifiers and the legend for the colors are shown in Fig. 4. While a detailed performance spectrum provides insight about individual cases, it does not directly visualize any quantified information. Therefore an aggregated performance spectrum serves for that purpose: within it, segments are split vertically into time windows, or bins, of a given duration, as shown in Fig. 2b).

Each bin contains a histogram that shows aggregated information about lines of the detailed performance spectrum that start, stop or intersect this bin. Besides the histograms, exact numbers are also available for users. Supported aggregation functions are presented in Fig. 3. In Fig. 2b) bars in bins show aggregation by cases pending function. For instance, line AB is counted within corresponding dark blue bars (i.e. for class 0-25%) in time windows tw1-tw3 of Z2. Additionally, parameter maximal observed throughput is shown within each segment (see Fig. 2b) (2)). It shows the maximal observed value of the aggregation function within bins of the segment. The required size of time windows, performance classifier and aggregation function are configured by a user before pre-processing of an event log.

**Fig. 3. Aggregation functions.**

<table>
<thead>
<tr>
<th>Aggregation function</th>
<th>Example</th>
<th>Result for bins</th>
</tr>
</thead>
<tbody>
<tr>
<td>cases pending</td>
<td><img src="example1.png" alt="Image" /></td>
<td>(1, 1, 1, 1)</td>
</tr>
<tr>
<td>cases started</td>
<td><img src="example2.png" alt="Image" /></td>
<td>(1, 0, 0, 0)</td>
</tr>
<tr>
<td>cases stopped</td>
<td><img src="example3.png" alt="Image" /></td>
<td>(0, 0, 0, 1)</td>
</tr>
</tbody>
</table>

**Fig. 4. Available in the PSM performance classifiers and their color codes.**

Within our tool, a user has a rich setool to explore a performance spectrum: 1) regular expression based filtering of segments by names, 2) filtering by throughput boundaries, 3) searching for traces in a performance spectrum by specifying their IDs, 4) providing various segment sorting orders. Additionally, a user can filter in particular performance classes, for instance, compare the spectrum in Fig. 3a), where only segments of classes 51-75% and 76-100% are shown, with the original spectrum in Fig. 2a). Another feature of the PSM allows to highlight all segments of cases that in the performance spectrum have lines that start in particular bins. For instance, in Fig. 3b) by selecting bin tw3 we highlight traces inside triangles ABC, CDE: they form a clearly distinguishable “hourglass” pattern within Z2-Z3, which shows that the traces are synchronized by activity Send Fine in point C. Interestingly, in Fig. 1 we observe more “hourglass”
patterns within Z2-Z3, together with other patterns, for example, strictly parallel lines of Z4 or spreading lines of Z6. Aforementioned features of the PSM allow to conduct extensive performance analysis of processes, including their performance patterns [4].

3 Extracting a Simpler Event Log and Main Behavior

In total, the event log contains 2,514,266 events for 43,809 applications over a period of three years. The shortest case contains 24 events, the longest 2,973 and on average there are 57 events per case referring to 14 activities. As expected, analysis of the whole log of that size leads to a “spaghetti” process model and too complex performance spectrum, so first we extract main behavior of the process, as recommended in [1]. In the following, we will filter the log in several steps to identify the top three variants of the main activities in the process (milestone activities) as shown in Fig. 6. To do this but not only analyze the main activities, we first create as intermediate result a log of the main activities and identify their top variants (L5), and then go back to the original log and select all cases and all events following these main variants (L6).

3.1 Identifying Sealed Activities

Analyzing the event log, we discovered that some activities can be successful or not (attribute Success is equal to true or false correspondingly). Interestingly, a part of such activities are always successful. The overview of such activities is shown in Fig. 7. Additionally, some activities are revocable, for example, decide can be revoked by revoke decision, begin payment can be aborted by abort payment. An activity can be revoked or aborted several times within one case and finally at most one non-revoked event of that activity is present, we will call these activity occurrences sealed. To separate sealed occurrences from cancelled ones, we renamed them by adding prefix “sealed” to activity names, e.g. sealed begin payment. Our schema of the event log simplification and extraction of main behavior is presented in Fig. 6. The log, enhanced with sealed activities, is shown as L2.

3.2 Identifying Milestone Activities

We formulate the following rule to identify milestone activities of the process: each milestone activity can occur only once in a case and each milestone activity should
**Fig. 6.** Pre-processing of the original event log.

**Fig. 7.** The observed options to revoke activities: 1) assigning attribute *Success* to *False*, 2) revoking by another activity.
be represented in almost all cases. In the log we found 17 such milestone activities: 

\texttt{Payment application-Application-mail income; Payment application-Application-mail valid; Parcel document-Main-initialize; Parcel document-Main-begin editing; Geo parcel document-Main-initialize; Control summary-Main-initialize; Control summary-Main-begin editing; Control summary-Main-finish editing; Reference alignment-Main-initialize; Reference alignment-Main-performed; Department control parcels-Main-performed; Entitlement application-Main-initialize; Payment application-Application-initialize; Entitlement application-Main-sealed decide; Payment application-Application-sealed decide; Payment application-Application-sealed begin payment; Payment application-Application-finish payment}. We then filtered the log to only contain these 17 milestone activities, resulting in \texttt{L3} of Fig. 6.

### 3.3 Aggregating Multistep Milestone Activities

Many milestone activities are presented in the event log with their life-cycle transitions, for instance, \texttt{Control summary-Main} has the following ones: \texttt{initialize, begin editing, finish editing}. They multiply the number of case variants, but not necessarily add information that is valuable for general understanding of the process. We aggregated activities that always have sequences of the same lifecycle transitions within a very short median interval of time (less than one second), choosing the timestamps of the first \texttt{initialize} event for the aggregated event, as shown in Fig. 8. We keep activity with \texttt{initialize} for consistency with names of the other milestone activities, it does not matter for analysis of the process. The resulting simplified event log with 13 milestone activities is shown as \texttt{L4} in Fig. 6.

![Fig. 8. Aggregating multistep activities into single-step ones.](image)

### 3.4 Validating Milestone Activities

Our general understanding of a milestone activity is that it should not have different order of execution in different cases. We verified this assumption using a log footprint [1] for 2017, built using the eventually follows relation (Fig. 9). Because the whole matrix is large, we provide an excerpt that contains the 13 milestone activities and several other activities. We can clearly see, that the majority of the milestone activities eventually follows other milestone activities, but only “one way”, demonstrating strict ordering. The only exception is \texttt{Payment Application-Application-mail valid}, which can occur before and after \texttt{mail income}. We explain that by errors in timestamps of activities \texttt{mail income}, which apparently are assigned by operators manually.
Fig. 9. The excerpt from the event log footprint for 2017: any milestone activity eventually follows others, but only “one way” (except the two first activities).
3.5 Extracting Main Behavior from the Simplified Event Log

In order to extract main behavior from L4, we identified most frequent case variants for each year separately in ProM, as shown in Fig. 10. The top 3 variants represent on average 81% of behavior. We kept only cases of those variants, and obtained the simplified event log which represents the main behavior of the process. It is shown as L5 in Fig. 6.

![Diagram]

Fig. 10. Top variants of 2015-2017 contain approximately 80% behavior.

3.6 Extracting Main Behavior from the Original Enhanced Event Log

To obtain an event log that represents main behavior but contains all the events, including ones with non-milestone activities, we filtered now L2 to contain only those cases of the three main variants contained in L5, thereby obtaining resulting event log L6. We extensively use obtained logs L5 and L6 for further analysis of the process.

3.7 Understanding the Process

The process documents, annual main steps and their relations are shown in Fig. 11. From the control-flow perspective, the process consists of two parts: the main sub-process and sub-processes for outliers. In the majority of cases the main sub-process includes the milestone activities we identified earlier, its process map is shown in Fig. 12. Outliers include cases, processed in sub-processes Payment application-Objection and Payment application-Change. Time-wise, cases start in the second quarter of each year and ideally should be closed by the end of year, i.e. sealed activity Payment application-Application-begin payment should be executed by the end of year. Late Payment application-Application-begin payment are considered as undesired outcome.

4 Performance Analysis and Concept Drift Detection

4.1 Performance Spectrum Parameters

In this paper we refer to many figures with fragments of the process performance spectrum. To describe such figures in a more clear and compact way, we provide names and values of parameters, which help to understand the figure. They are presented in
Fig. 11. Documents and main steps of the process during a year.

Fig. 13 together with their short notation. The majority of parameters are self-explanatory, but some of them require explanation: timestamp shifting describes how timestamps are changed to make comparative visual analysis easier, segment factory describes how segments are defined: using directly/eventually follows relation (DFR/EFR), or occurrences of activities (1-Activity). For instance, the caption of Fig. 14(c) means that the aggregated performance spectrum is built from the original event log with “Document-Subprocess-Activity” activity classifier, which is pre-processed by 1) adding information about sealed activities, 2) projecting (filtering in) the milestone activities, 3) filtering in 80% behavior; the spectrum is built using the directly follows relation to generate its segments, bins with duration of one week; segments are sorted by the order of the milestone activities. The caption does not describe an aggregation function and performance classifier, because in this report, by default, all instances of the performance spectrum are built using the quartile-based performance classifier and aggregation function cases started (see App. A).

4.2 General Performance Spectrum

First we obtain general insight into the process performance. For that we import simplified event log L5 into the PSM and sort the segments according to the model in Fig. 12. The result is shown in Fig. 14 where we can easily distinguish three one-year periods of activity and gaps in between. By intuition, each period contains only traces of the corresponding year, i.e. where trace attribute Year is assigned to the corresponding year (2015-2017). To verify that, we filter case IDs of each year and highlighted them in the performance spectrum (using feature highlighting of traces by their IDs). Thus we conclude that any stage of the process has cases for a period of 1-2 months, but not longer and all cases gradually move together through the stages.

Activities across Years In Fig. 14(a) we can observe that over years, as stated in the assignment, from 2015 to 2016, the Parcel document was succeeded by the Geo Parcel Document. In 2017, the Geo Parcel document also replaced the Department Control Parcels document. Interestingly, in 2016-2017 Entitlement application-related activities are still presented in the process (L4) but no more presented in main behavior (L5).

4.3 Understanding Performance Patterns

In this section we explain how to apply the concept of the performance spectrum and performance patterns of processes by example. As an example we selected a part of the
Fig. 12. The process map: the main behavior in 2015-2017.
Fig. 13. Performance spectrum parameters.

A performance spectrum that corresponds to the first three segments S1-S3 of the top variant of 2017 and the period between 06.04.17 and 18.05.17, because it contains the most diverse performance spectrum within simplified event log L5. The corresponding performance spectrum is presented in Fig. 15. Each cell in the grid frames one calendar day in the CET time. The top part of the figure contains the detailed performance spectrum, the bottom one contains the same spectrum with some lines highlighted. The aggregated performance spectrum calculated using function cases started. In our case for S1-S3, a number of lines, started within a time window (bin) of a segment, is equal to a number of lines stopped in the same time window of the previous segments, because only one case variant is presented. The highlighted lines correspond to cases that have lines started in bins w1-w2; they are shown in colors and solid lines, while others are shown in gray color and dotted lines. The both parts are required to explain the performance.

Within the reasoning given next, we can observe in Fig. 15 that mails are validated at various speed without an order. In segment S1 lines start every day, including weekends and holidays, around midnight, and stop every day, excluding weekends and holidays, such as Friday before and Monday after the Easter Sunday, the Labour Day, also around midnight. Apparently farmers’ mails trigger the first activity of the process, Payment application-Application-mail income, every calendar day, but in the system such mails are registered (the same day or later) with the one-day precision. The second activity of S1, Payment application-Application-mail valid, occurs only in working days, so we can guess that it is a manual activity, which is registered with the one-day precision. Based on the aggregated performance spectrum, we observe that the number of starting and stopping lines within S1 grows toward the end of the period, with the highest peak of starts on May 10 in bin P1, and the highest peak of stops on XXX in bin P2. Are the lines of S1 ordered? To understand that, we selected bins w1, w2, which correspond to peaks of stopped lines in S1, and obtained two triangles ABN, ACF in S1, which are formed by lines that stop in w1, w2 (see Fig. 15b). The lines of the triangles ABN, ACF cross each other within triangle ABT, thereby demonstrating unordered execution.
Fig. 14. The performance spectrum of event log LOG(DSA, milestones, 80%, sealed). a) activities across years [SPECTR(1A, 1w), VIEW(SortBy(milestones))]. b-c) The detailed and aggregated performance spectrum [SPECTR(DFR, 1w, Name Aggr.(renamed)), VIEW(SortBy(milestones))].
Fig. 15. Detecting performance patterns: the detailed (a) and combined (b) performance spectrum of segments S1-S3 with highlighted performance patterns. ACFD = [Scope(seg,glob), Shape(comb,unord), Work(cont, grow, cases start), Perf(25%,4 classes)], DEKH, DGL = [Scope(seg,overlapp), Shape(comb,batch(e)), Work(sparse, grow, cases start), Perf(25%,4 classes)], r1-r2 = [Scope(seg,arb), Shape(comb,batch(s,e)), Work(sparse, grow, cases start), Perf(25%,1 class)]
of corresponding cases. After we repeated that for other bins, we concluded that segment S1 within the presented period of time, i.e., inside rectangle ACFD, contains many traces of variable duration, which are continuously distributed over time and can overtake each other. In terms of the performance patterns of App. \[ A \] \[
\text{ACFD} = [\text{Scope(seg,glob)}, \text{Shape(comb,unord)}, \text{Work(cont, grow, cases start)}, \text{Perf}(25\%, 4 \text{ classes})].
\]

In segment S2 lines start exactly as the lines of S1 stop, but stop within two short periods of time, which are highlighted by rectangles r1, r2. r1 has duration of 5 days, including a weekend, and r1 has duration of 1 working day. These two periods of execution form batching processing on end, which can be describe as DEKH, DGL = [Scope(seg,overlapp), Shape(comb,batch(e)), Work(sparse, grow, cases start), Perf(25\%, 4 \text{ classes})]. We observe that cases are synchronized by activity Geo Parcel document-Main-initialize within periods r1 and r2, with the peak in r2.

In segment S3 we observe parallel lines inside two rectangles r1, r2, which we can describe as batching processing on start and end: r1, r2 = [Scope(seg,arb), Shape(comb, batch(s,e)), Work(sparse, grow, cases start), Perf(25\%,1 \text{ class})]. We observe that after activities Geo Parcel document-Main-initialize Control Summary-Main-initialize is instantly executed.

A possible schematic representation of the patterns of S1-S3 is presented in Fig. 17 (a). This representation omits classification of duration, exact scale and workload, but preserve many main characteristics. We will use this characterization in Sect. 4.5 when discussing the general performance of the process.

### 4.4 Detecting Concept Drift in Performance Spectrum

To demonstrate further the approach, explained in Sect. 4.3 and start comparative analysis of the process performance across years, we described performance patterns of the first three segments also for 2016-2017, as shown in Fig. 17 (b-c). To directly compare the processes and detect concept drift in performance, we shifted the cases of 2016 and 2017 by -1 and -2 years respectively, and generated the performance spectra for S1-S3 for all three years one under another as shown in Fig. 16. In the first segment Payment application-Application-mail income:Payment application-Application-mail valid the performance pattern is similar for each year, it is described Sect. 4.3 The only difference is in dates when the cases start: 10.04 in 2015, 04.04 in 2016 and 27.03 in 2017. Each next year the process starts one week earlier and the period between Payment application-Application-mail income and Payment application-Application-mail valid lasts one week longer, thereby allowing a more steady pace for incoming cases.

More interesting patterns we observe in the second segment Payment application-Application-mail valid: (Geo) Parcel document-Main-initialize in Fig. 17 in 2015-2016 we observe single instances of batching: P2.1-2 = [Scope(seg.once), Shape(comb,batch(e)), Work(cont, grow, cases start), Perf(25\%,4 \text{ classes})], while in 2017 we observe two instances of batching: P2.3 = [Scope(seg,overlapp), Shape(comb,batch(e)), Work(sparse, grow, cases start), Perf(25\%, 4 \text{ classes})]. More instances of batch processing split the total workload, making the processing between Payment application-Application-mail valid (Geo) Parcel document-Main-initialize more homogeneous in 2017.
The third segment (Geo) Parcel document-Main-initialize:Control summary-Main-initialize shows different patterns each year: P3.1 = [Scope(seg,once), Shape(comb, batch(s,e)), Work(sparse, fall, cases start), Perf(25%, 1 class)] in 2015, two different patterns P3.2 = [Scope(seg,once), Shape(comb, batch(s,e)), Work(sparse, fall, cases start), Perf(25%, 1 fast class)] and P4.2 = [Scope(seg,once), Shape(comb, batch(s,e)), Work(sparse, peak, cases start), Perf(25%, 1 slow class)] in 2016 and P3.3 = [Scope(seg,arb), Shape(comb, batch(s,e)), Work(sparse, grow, cases start), Perf(25%, 1 class)] in 2017. In those patterns we observe the following behavior. In 2015 Parcel document-Main-initialize start as soon as Payment application-Application-mail valid stop being executed and processed during three working days, generating a high peak of workload. In 2016 we observe similar behavior except one more batching processing, which starts together with the first one but lasts longer, approximately three weeks, thereby making the processing more homogeneous but also increasing maximal duration between Payment application-Application-mail valid and Control summary-Main-initialize. In 2017 again two batching instances are presented, but they complete withing the time frame of the previous segment, thereby not only making the process more homogeneous, but also shortening maximal duration between Payment application-Application-mail valid and Control summary-Main-initialize.

4.5 General Process and Process Changes in Terms of Performance Patterns

By applying the approach, demonstrated in Sect. 4.3-4.4, we described the performance patterns for other segments of L5. It is presented in Fig. 18, where the detailed spectrum is drawn using different shapes for different patterns, as shown in the legend; the shapes are annotated with % of workload and duration between the most left and right point of each shape. Besides performance aspects described in Sect. 4.4 we found out the following differences across years:

- maximal duration between beginning of cases and Control summary-Main-initialize significantly decreases in 2017;
- various performance variants appear and disappear over years;
- distribution of workload between two performance variants Geo Parcel document-Main-initialize and Reference alignment-Main-initialize is different for 2016 and 2017: 46% and 85% correspondingly;
- duration between Payment application-Application-sealed begin payment and Payment application-Application-finish payment in 2016-2017 is three times shorter than in 2015.

4.6 Exploring Undesired Outcome 1

According to the assignment, undesired outcome 1 is defined as the payment is late: a payment can be considered timely, if there has been a “begin payment” activity by the end of the year that was not eventually followed by “abort payment”.

In order to explore such cases, we build the performance spectrum for the milestone activities in 2015-2016 (Fig. 19 a-b). We exclude 2017 from this analysis, because the
Fig. 16. The combined performance spectrum of segments S1-S3 for 2015-2017; [LOG(DSA, milestone, 80%, sealed, TimeShift(2015)), SPECTR(DFR, 1d), VIEW(SortBy(milestones))]
Fig. 17. The performance of S1-S3 in terms of the performance patterns (see App. A).

Fig. 18. The process performance in terms of the performance patterns (see App. A). *Duration shows periods between the most left and right points of each shape.
original event log does not contain complete information for 2018. To highlight cases
with undesired outcome 1, we select cases, that presented in S14 (Payment application-
Application-begin payment:Payment application-Application-finish payment) after the
end of the year (inside the dashed rectangles). In Fig. 19, the colored solid lines of the
detailed performance spectrum belong to such cases, while dashed grey lines belong to
others. In the resulting performance spectrum we observe the following:

- the numbers of cases of undesired outcome 1 are slightly different: 96 (2015) and 72
  (2016);
- from the beginning of the process until Entitlement application-Main-initialize(2015)
  and Payment application-Application-sealed decide(2016) all the cases follow the
  same control flow and performance variant;
- the majority of cases of undesired outcome 1 are “late” because of activities Entitle-
  ment application-Main-sealed decide and Payment application-Application-sealed
decide(2015), and Payment application-Application-sealed begin payment(2016);
- in both years cases of undesired outcome 1 last for many months of the next year.

So we can conclude that “responsibility” for undesired outcome 1 shifted from activ-
ities Entitlement application-Main-sealed decide and Payment application-Application-
sealed decide in 2015 to Payment application-Application-sealed begin payment in 2016,
without significant change in the overall number of such cases.

4.7 Typos in Timestamps
According to the assignment, some timestamps are manually entered and may therefore
contain spelling mistakes. So we decided to find out if it is possible to detect such errors
in the performance spectrum. Exploring the detailed performance spectrum, built by
L6, we found “suspicious” cases that have activity of the process beginning in one year
and others in the next year. For example, in Fig. 20 a) one of segments Entitlement
application-Main-mail valid: Payment application-Application-mail income: starts in
2014 and ends in 2015. Using feature “highlighting traces that start within selected bins”,
we selected this segment, as shown in Fig. 20 (a), thereby highlighting the corresponding
case, which shown in Fig. 20 (b). While selecting, the corresponding case ID was copied
onto the clipboard: 980452d16c89c2c3. This only case has the following sequence
of years of events timestamps at the beginning: 2014, 2015, 2014, 2014, 2015... and
attribute Year = 2015 in the original event log. We concluded, that timestamps of those
activities look as if an operator made typos. Totally we found IDs of only six cases
with wrong timestamps: 50f57bfbcac17b65; cedcde09b55a5da7; 13bd492abca27e38;
2e0f33f7ae3fadd9; c3692d02d388ee2a; 0d18b291460d4aa0. Despite the presence of six
errors in timestamps is not critical for our analysis, in general a performance spectrum
can help to easily detect similar types of outliers.

4.8 Parcel Document and Geo Parcel Document: Changes in Individual Steps of
the Process
As we know from the assignment, from 2015 to 2016, the Parcel Document was suc-
cceeded by the Geo Parcel Document. What are the changes in individual steps of the
Fig. 19. Cases of undesired outcome 1 (i.e., with late Payment application-Application-begin payment) within segments S1-S14 (see Fig. 12) are shown in solid lines for 2015 (a) and 2016 (b); [LOG(DSA, milestone, sealed), SPECTR(DFR, Name aggr.(renamed)), VIEW(SortBy(milestones))]}
Fig. 20. Identifying cases with wrong timestamps in the detailed performance spectrum: case 980452d16c80c2c3 belongs to 2015 but surprisingly contains events for 2014. To answer that, we built the aggregated performance spectrum of segments with activities (Geo) Parcel Document and aggregated them using “Activity to Any” aggregation. The resulting spectrum is shown in Fig. 21. In 2016 the following changes happened:

- sub-process Declared was added;
- activities begin editing and finish editing were moved from sub-process Main to Declared;
- three new activities were added to sub-process Declared: create, calculate, restart editing.

In 2017 the following changes happened:

- activity restart editing disappeared;
- two new activities were added to sub-process Declared: finish pre-check, clear;
- new sub-process Reported was added;
- in new sub-process Reported activities finish editing stared to occur before others, while other activities of this sub-process stared to occur only from 4th quarter of 2017;
- new activity calculate protocol was added for cases of 2016 and 2017.

To analyze performance those sub-processes in more detail, we additionally explore the combined performance spectrum of the same sub-processes. In Fig. 22 the combined performance spectrum of cases that belong to 2015 is highlighted. Interestingly, we still observe activities of the replaced in 2015 Parcel Document sub-process in 2016-2017 for cases that belong to 2015.

Exploring the performance spectrum, we could easily describe how documents, sub-processes, activities and steps of the sub-processes, represented as segments, appear and disappear over years, including quantified information about numbers of corresponding cases. Also we detected a possible anomaly in new sub-process Reported.

4.9 Dynamics of Revocable Activities

In Sect. 3.1 we identified activities that can be revoked or aborted, and enhanced the original event log with information about sealed activities, i.e. final activities that were
Fig. 21. Comparative analysis of sub-processes for *Parcel Document* and *Geo Parcel Document*; [LOG(DSA, Act.Proj.(contains("parcel document"))), SPECTR(1A, 1w), VIEW(SortBy(custom), Thr.Filt.(> 100))]

Fig. 22. *Parcel Document* over years: despite replacing *Parcel Document* with *Geo Parcel Document* in 2015, we still observe its activities in 2016-2017 for cases that belong to 2015.
not revoked or aborted. Because cancellation of such activities implies that extra work has been done for a case, that may be a subject of interest for the process owner, so we analyzed such scenarios in more detail.

**Analysis of Aborted “Begin Payment” Activities over Years** In this section we analyze activities `begin payment`, which can be aborted, in order to understand how efficiently they were processed in 2015-2017. For that, we obtained the aggregated performance spectrum (Fig. 23), built from L6. In the spectrum we can clearly see that `begin payment` can be aborted with sub-processes `Payment application-Application/Objection/Change`. In the scope of this report, we focused only on sub-process `Payment application-Application`, in order to demonstrate the technique. To distinguish cases of different years in the performance spectrum, we filtered case IDs for required years in Disco and highlighted them in the PSM. For the chosen sub-process, we concluded the following:

- all sealed activities `begin payment` were executed during a short interval in the middle of December for all cases;
- in 2015 in 100% of cases activity `begin payment` was aborted one or more times;
- in 2016 in 71% of cases activity `begin payment` was aborted one or more times;
- in 2017 in 13% of cases activity `begin payment` was aborted one or more times.

Therefore over years for sub-process `Payment application-Application` efficiency of activities, related to beginning of payment, dramatically increased, by reducing the amount of extra work related to repeating aborts of payments.

![Fig. 23. Activities abort payment over years: 100% cases in 2015, 71% in 2016 and 13% in 2017; LOG(DSA, Act.Proj.(ends("payment")), SPECTR(1A, 1w), VIEW(SortBy(custom)))](image)

**Analysis of Revoked “Decide” Activities over Years** Similar to the previous section, in this section we analyze efficiency of decision making, exploring activities `decide` of sub-processes `Entitlement application-Main` and `Payment application-Application`, which can be revoked by activity `revoked decision`. For that we filtered in all activities that end with `sealed decide` or `revoke decision and` where at least one activity is a milestone, and obtained the aggregated performance spectrum, built from L6. In Fig. 24 we can observe, that `Payment Application-Application-Decide` was canceled in 38% of cases in 2015, in 100% of cases in 2016 and in 2% of cases in 2017. We can conclude that a number of revoked cases reached its peak of 100% cases in 2016 but was drastically decreased to almost zero in 2017, thereby increasing efficiency of those activities.
Remote Inspection. So far we analyzed the process disregarding resources that execute its activities. In order to demonstrate how to work with the organizational perspective, we select sub-process Inspection-Remote for our analysis and build its process map (Fig. 25). As shown in the process submap of the sub-process Inspection-Remote (Fig. 25), the activities of the sub-process are mostly executed by three resources: Inspection automation, Remote inspection export, and Remote inspection import. To explore handover of work between the most frequent resources Remote inspection export and Remote inspection import, we also build the performance spectrum, using classifier resource-department Any:A and aggregation of activities Any:A (Fig. 26). In the performance spectrum, we can observe the following:

- By 2017 handover of work to resources Remote inspection export/import becomes more homogeneous with a shorter (than in 2016) period of time between activities related to export and import.

- Other resources handover work to Remote inspection export during one short period in 2015-2016 and during two short periods in 2017.

- Other resources handover work to Remote inspection import during one short period in 2015 and during two short periods in 2016.

- The interval between export- and import-related activities grows in 2016 and decreases in 2017.

- Resource Remote inspection export has less work from department 4e in 2015, 6b in 2016 and 6b in 2017.

- Resource Remote inspection import has less work from department 4e in 2015, 6b in 2016 and 6b in 2017.

- The interval between export- and import-related activities grows in 2016 and decreases in 2017.

- Resource Remote inspection export has less work from department 4e in 2015, 6b in 2016 and 6b in 2017.

- Resource Remote inspection import has less work from department 4e in 2015 and 6b in 2016.
Payment Application-Application-Abort Payment  As another example of performance analysis of the organizational perspective, we selected processing of activity Payment application-Application-abort payment. In the combined performance spectrum, presented in Fig. 27, we can observe the following:

– all the departments process activity Payment application-Application-abort payment in 2015;
– all the departments except e7 process activity Payment application-Application-abort payment in 2016;
– in 2015 department e7 processes two times more Payment application-Application-abort payment than other departments. Analyzing case frequencies, we find out that that happens because of repeating Payment application-Application-abort payment in cases of this department;
– in 2017 department d4 processes almost all Payment application-Application-abort payment.

Analyzing the spectrum of 2015-2016 we can guess that department e7 do not process activity Payment application-Application-abort payment in 2016, because its resources were cause of the double number of abort payment in 2015.
5 The General Approach of Work with Performance Spectra of Processes

So far we apply our technique to the concrete problems of this challenge. In this section we summarize steps, commonly required to analyze performance of processes using our approach, as follows:

1. get first insight about the process by exploring a performance spectrum of the whole log, use aggregation to obtain a more compact and clear visualization;
2. simplify the original event log and extract main behavior;
3. discover a control-flow model;
4. build the performance spectrum of the obtained event log, sort its segments using the control-flow model;
5. analyze the obtained performance spectrum, detect and describe performance variants and patterns within each segment, provide interpretation of findings;
6. repeat the previous step for larger parts of the spectrum, e.g. for two and more segments;
7. pre-process the original event log in a different way to extract other kinds of information, for example, behavior of outliers, sub-processes without simplification, other periods of time, the organizational perspective and so on, and repeat steps 3-6;
8. summarize obtained findings.

6 Conclusion

In this report we applied the recently developed performance spectrum to analyze concept drift in the control-flow and in the performance perspective of the BPI Challenge 2018 event log. After preprocessing the log to contain the top 3 process variants in each year, we conducted an exploratory analysis of which we distilled lessons learned and best practices for conducting an analysis in the performance spectrum. Following these practices, we obtained the following findings.

1. Through the performance spectrum, we found that all cases of the process run largely parallel throughout the year, all cases progressing through multiple stages taking 1-2 months time, and stages are concluded by all cases passing through a certain milestone activity as a batch. Within this general characteristics, the process changes over the years both in control-flow, but most notably in its performance characteristics.
2. As major control-flow change, we could identify the replacement of parcel by geo-parcel documents in the performance spectrum as a pattern where segments related to geo-parcel documents in years 2016 and 2017 showed behavior during the same stages of the process where in 2015 segments related to parcel documents showed behavior.
3. This change in moving from parcel to geo-parcel documents introduces a new performance variant into the process in 2015 for steps after the (geo) parcel document: in the new variant cases reach the control summary and the payment application later (with longer lead times), and the share of cases going through these new variants...
grows from 46% and 87% in 2016 to 85% and 100% in 2017, respectively. This change in performance however has no impact on the number of undesired outcome of cases (late payments). The detailed performance spectrum however clearly highlights the cause of late payments which has shifts to a different activity.

4. Also coinciding with the introduction of geo-parcel documents is a significant decrease in time until the control-summary documented is reached due to a change in the hand-over from payment applications to (geo-)parcel documents: in 2015, we see a strong LIFO behavior where later cases are processed before earlier cases, whereas 2016 introduces batching of cases and 2017 finally shows several batching moments at this hand-over leading to a reduction of lead times.

5. Further, through the aggregated performance spectrum shows that the share of cases where payments were aborted (and then had to be started again) dropped from 100% in 2015 to 13% in 2017. Similarly, decision activities that had to be revoked and done again dropped from 38% in 2015 to 2% in 2017 (though in 2016 all decisions had to be done again). Altogether, this suggests that the changes made in the earlier stages of the process have a significant positive impact on the later stages, though the adaptation causes some intermediate problems in 2016.

6. We also showed that the performance spectrum aids in findings changes in the resource perspective. In 2015, payments applications by four different departments had to be aborted, with one department having twice as many aborted applications, whereas in 2017 only payment applications from one department had to be aborted.

Altogether, we could demonstrate that the performance spectrum and its implementation is a novel, yet mature concept for identifying and analyzing changes and concept drift in the control-flow and especially in the performance perspective of business processes. It allows to reveal and visualize details in process performance that cannot be obtained with other existing techniques. The richness in details, however, comes at a cost of careful methodological application. We consider the summarized lessons learned in applying the technique to further mature the application of the performance spectrum in process analysis.

Appendices

A Taxonomy of Elementary Patterns

In this Appendix we provide an excerpt from from [4] with the taxonomy of elementary patterns, in order to make this report self-contained.

We observed a great variety of elementary patterns and combinations of patterns in the performance spectra of real-life processes [4]. That makes it impossible to provide a comprehensive catalog. Nevertheless, we are able to provide a comprehensive taxonomy of parameters of elementary patterns. It allows us to completely and unambiguously describe performance of a process over time in a way that patterns that correspond to similar performance scenarios have identical descriptions and identical descriptions of patterns mean similar performance scenarios, while changing the value of any parameter in a pattern would mean a different performance scenario.
The taxonomy provides parameters to characterize the Shape of lines and bars in a process in a particular Scope over time; line density and bar height describe Workload while their color describes Performance. The parameter values form a hierarchy which is shown together with typical patterns having these characteristics in Fig. 28. We provide a unique short-hand value [in brackets] for each parameter, to allow succinct notation of patterns.

**Scope parameters** capture the place of pattern in the performance spectrum.

- **size**: one segment [1 seg], one subsequence [1 sub-seq], several subsequences [>1 sub-seq]
- **occurrence**: globally [glob], as a local instance [loc]
- **repetitions** (for patterns occurring in local instance): once [once], regular [reg], periodic [per=T], arbitrary [arb],
- **overlap** (for repeating patterns): overlapping [overlap], non-overlapping
- **duration**: absolute value [D=T]
Size describes the pattern length from the control-flow perspective: a single segment, a single subsequence or several subsequences of event classifiers. Although all elementary patterns have size 1 seg, we include this parameter in the taxonomy for compatibility with composite patterns. A pattern occurrence can be either global, when it occurs continuously throughout a segment without clear boundaries, otherwise it distinctly occurs as a local instance. Pattern instances may occur once or repeat (1) periodically in particular intervals $T$, (2) regularly, i.e., seemingly systematic but not periodic, or (3) arbitrarily. Repeated pattern instances can be overlapping or non-overlapping in time. Parameter duration describes the absolute duration over time (e.g. as an interval in seconds).

**Shape parameters** describe the appearance of lines and bars in the visualization of the performance spectrum.

- **type**: detailed [det], aggregated [agg], combined [comb]
- **order**: unordered [unord], LIFO [LIFO], FIFO with variable time [FIFO-var], FIFO with constant time [FIFO-const], batching on start [batch(s)], batching on end [batch(e)], batching on start and end [batch(s+e)]

A pattern described just in terms of lines (bars) of a detailed (aggregated) performance spectrum is detailed (aggregated); if it requires both it is combined. Order describes the configuration of lines in a detailed pattern: (1) unordered when lines irregularly cross each other, (2) LIFO when lines end in reversed order of starting, (3) FIFO when lines never cross. (3b) Non-crossing lines of variable inclination mean variable time [FIFO-var], where multiple lines starting (or ending) in a very short period show multiple cases batching on start (or on end). (3c) Lines of identical inclination show constant time [FIFO-const], where multiple lines starting and ending in a very short period (with no lines before/after) show batching on start and end.

**Workload** describes the height of bars in aggregated or combined patterns, and the density of lines in detailed patterns over time.

- **aggregation function**: segment instances started [start], stopped [stop], cases pending [pend]
- **workload character**: continuous [cont], sparse [sparse]
- **amount of workload**: zero [0], non-zero [>0], low [low], medium [med], high [high]
- **workload trends** (for a performance class or in total): can be steady [steady], variable [var], growing [grows], falling [falls], showing peaks [peak] or drops [drop]

Workload is characterized by the aggregation function defined in the view of the performance spectrum. Workload character can be continuous or sparse (when there are longer gaps between lines or bars), and it is visible in both detailed and aggregated patterns. Amount of workload is categorized as zero or non-zero, the latter can be categorized further as low, medium or high in relation to the maximum number of observations made on a segment (within the time period $p$ of the view). The trend over time can be steady (bars have about same height) or variable, the latter splits further into steadily growing, falling workload or showing peaks (a few high bars surrounded by lower bars) or drops.
Performance is described in terms of the performance classes present in the pattern with respect to the classifier $C$ of the view chosen by the user.

- **Classes presented**: $1, > 1$, number of classes, subset of classes
- **Classifiers**: various, we discuss quartile-based [25%] (e.g., all observations belonging to the 26%-50% quartile), median-proportional [x·med] (e.g., all observations 2-3 times longer than the median duration)

In the visualization of the performance spectrum, classes are coded by colors. A monochrome pattern has 1 class presented while a multi-colored one has > 1 classes presented.

Now we show how the taxonomy describes the elementary patterns E1-E3 found in the RF log and highlighted in Fig. 29(left). Pattern E1 occurs in a single segment in local instances with a duration of 6 months, instances repeat regularly and overlap; the detailed pattern shows batching on end in a continuous workload for 4 performance classes in a quartile-based classifier. Using the short-hand notation, we write $E1 = \{\text{Scope(seg,loc,reg,overlap,D=6mo)}, \text{Shape(det,batch(e))}, \text{Work(cont)}, \text{Perf(25%,4 classes)}\}$. Similarly, we can characterize $E2 = \{\text{Scope(seg,glob)}, \text{Shape(det, FIFO-const)}, \text{Work(sparse)}, \text{Perf(25%, 1 class)}\}$ and $E3 = \{\text{Scope(seg,loc,reg, overlap, D=1mo)}, \text{Shape(det, batch(s))}, \text{Wo(cont)}, \text{Perf(25%, 4 classes)}\}$.

In case of creating a catalog of elementary patterns, some additional information can be added to pattern descriptions: a unique identifier and name and a meaning depending on the domain and the chosen event classifier, e.g., resources in a business process, or physical locations of a material handling system.

### References