

Analysis templates for identifying improvement opportunities using Apromore

Supplementary Material B

Nadiia Danylyshyn

Table of contents

Introduction	4
Activity-related improvement opportunities	7
Small activities	7
Large activities	10
Activity variants	12
Similar activities	16
Resource-related improvement opportunities	18
High/low resource utilization	18
Activity- and resource-related improvement opportunities.....	22
Controls performed by internal resources	22
Independent sequential activities.....	25
Frequent handovers.....	28
Ping pong behavior	31
Control flow-related improvement opportunities	35
Rework.....	35
Knock-out	38
Workaround	42
Waiting time-related improvement opportunities.....	45
Highest waiting time in the business process	45

Cases with the highest waiting times.....	50
Bottleneck.....	55
Manual time-consuming fragment.....	59
Process-related improvement opportunities	62
Manual process	62
High process complexity	65
Similar process variants.....	67
Process wastes-related improvement opportunities.....	71
Overprocessing	71
Overproduction.....	74
List of references	76

Introduction

Continuous business process improvement is an essential part of process management. Organizations that aim to make their processes more efficient, need to analyze processes to detect and address improvement opportunities on a rolling basis. The threat of neglecting process improvements is decrease of process performance. A business process improvement opportunity can be defined as a pattern in the process that has the potential to be improved. To address the improvement opportunity, process redesign is applied.

The current document aims to provide a guideline to find different improvement opportunities in the form of templates. Templates are tailored mainly for people with basic to good knowledge of BPM and elementary to intermediate skills with Apromore. Step-by-step guidelines are compiled based on six data sources: 2011-2020 BPIC submission reports, Lashkevich and Milani [1], Sharma [2], Reijers and Mansar [3], Dumas [44] and Dumas [45]; redesign possibilities are based on Lashkevich and Milani [1]. All the instructions are written for identifying improvement opportunities using Apromore specifically, but the described approaches can be applied to other process mining tools.

This work covers 21 improvement opportunities presented in the form of templates. Each template has the same structure:

- **Improvement opportunity (IO)** – name of the improvement opportunity
- **Definition** – short description of the improvement opportunity
- **Examples** – short scenarios that demonstrate the improvement opportunity
- **Minimum data needed** – list of minimum data in the event log (e.g., activities, end timestamps) that are required to identify improvement opportunity
- **Guideline on how to identify this IO** – step-by-step instruction on how to identify improvement opportunity in the process. The section starts with expected output that describes the main findings after completing certain step(s). It is followed by the instruction itself, which consists of step description (what you should do), Apromore example (screenshot with example of the output after completing the step), and explanation (some hints of what you should look for in the step and the description of the result of the step). The last part is output which describes the overall result after completing all the steps. This section allows to find the improvement opportunity but not the reasons why this improvement opportunity occurred.
- **Redesign possibilities** – generic ideas on how processes can be redesigned to address the improvement opportunity.

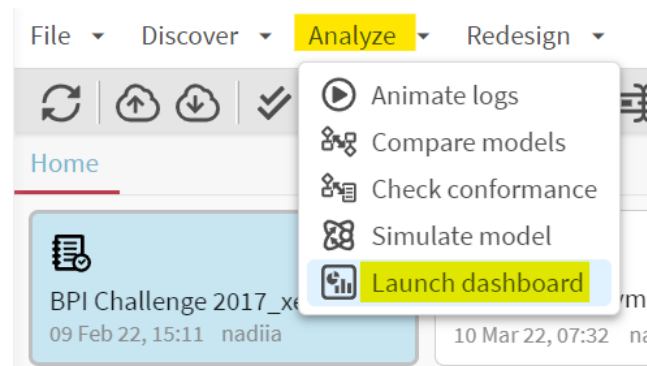
- **References** – list of references. This covers references to academic literature and/or references to BPIC submissions. Academic papers can cover case studies that address improvement opportunity, description of new method on how to define the improvement opportunity, description of the improvement opportunity. References to BPIC submissions provide articles where the particular improvement opportunity was identified with a short description of the process. References are not Apromore-specific, they depict improvement opportunity either for some other process mining tool (e.g., Celonis, ProM, Disco) or without linkage to particular process mining tool.

How to use templates?

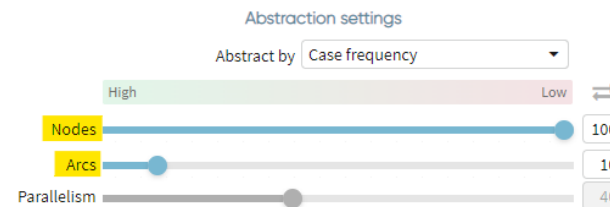
- The data for analysis focused on opportunities for process improvement identification should be pre-processed as this phase is not covered in the templates
- Based on the needs of your analysis, you can either check all the opportunities presented or only ones you are specifically interested in
- Before identifying improvement opportunity in the event log (using “Guideline on how to define this IO” section), read the whole template to better understand the improvement opportunity
- As most of the steps require listing some elements, it is recommended to have a pen and paper or document file for taking notes
- Many of the templates will ask to most/less frequent activities, highest/lowest waiting times or similar. In such cases, please define the threshold based on the needs of your analysis

Some tips about using Apromore [4]:

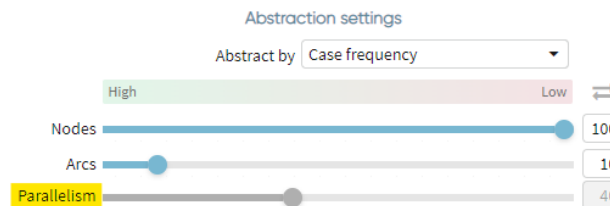
- In order to open an event log in the process discoverer, double-click on the file in Apromore Portal.
- In order to open the dashboard, select event log in the Apromore Portal, click *Analyze* and choose *Launch dashboard*:



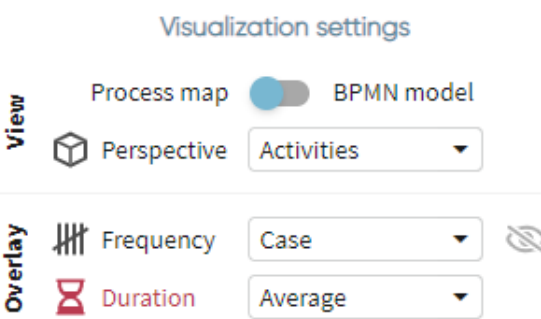
- By default, the generated process map has 100% of nodes (activities) and 10% of arcs (direct relations between activities). You can adjust the percentage of visible arcs and nodes using corresponding sliders of Abstraction settings. When increasing nodes and arcs, Apromore gradually adds less frequent elements. If nodes are set to 0%, only the most frequent activities are displayed, when it is 100%, all the nodes are visible. If arcs are at 0%, only the main relations between activities are presented, when it is 100%, all the arcs are presented.



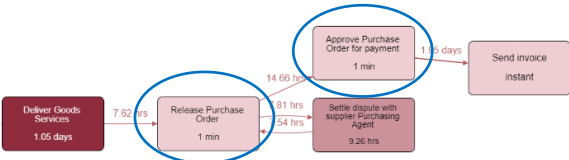
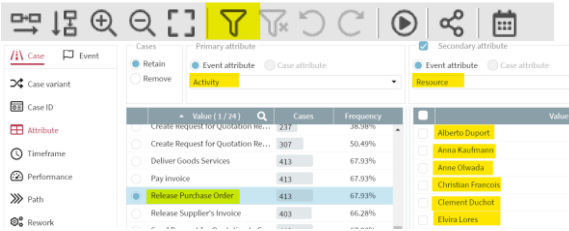
- For generated BPMN models, it is possible to adjust the percentage of visible parallel gateways using Parallelism slider of Abstraction settings. If parallelism is set to 0%, no parallel gateways will be displayed, if it is 100, all the parallel gateways will be presented.



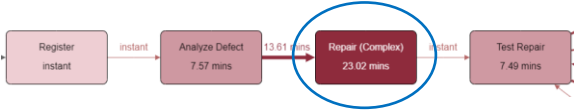
Activity-related improvement opportunities

1. Improvement opportunity (IO)	Small activities		
2. Definition	An activity that has one or several functional procedures and short processing time		
3. Examples	In the pharmacy process: <ul style="list-style-type: none"> - Activity “Notify a client that medicines are ready” is a small activity because it takes a short time to perform and a few functional procedures 		
4. Minimum data needed	Activities, start timestamps, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: <i>Steps 1-2</i> allow determining the list of small activities. <i>Step 3</i> checks if these activities are performed by human resources.		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the <i>Visualization settings</i> , select <i>Duration overlay</i> and choose <i>Average</i> . In the <i>View</i> section, choose the <i>Activities</i> perspective	

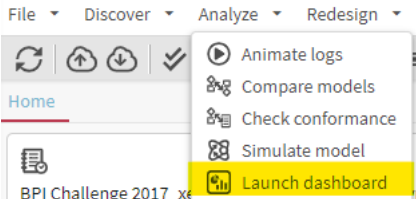


Result of the step: generated process map based on average duration and activities perspective

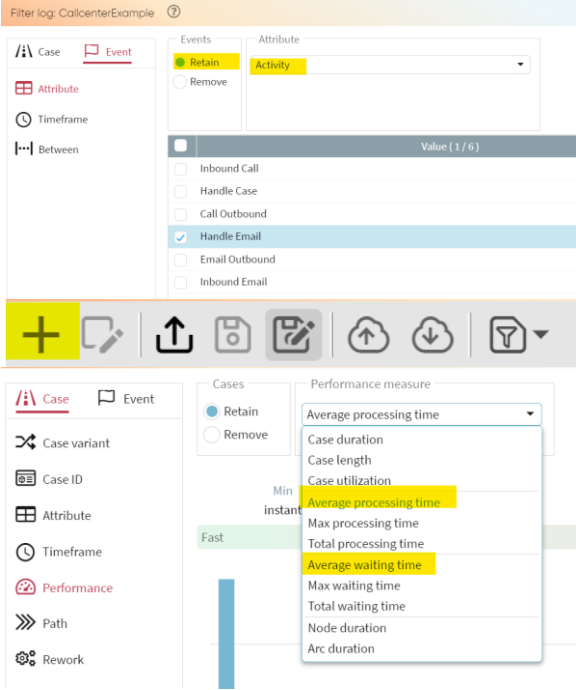

	2	<p>From the process map, find activities that have the shortest processing time but not instant. List these activities</p>		<p>The lighter the activity is the shorter is the processing time. On the example screenshot, small activity is circled in blue</p> <p>Do not include instant activities because most probably they are executed by system or other non-human resource and are impossible to redesign</p> <p>Result of the step: list of small activities</p>
	3	<p>Open the <i>Filter log</i> screen. In Case → <i>Attribute</i> filter, set Primary attribute as <i>Event attribute, Activity</i>; set Secondary attribute as <i>Event attribute, Resource</i>. Select an activity identified in <i>Step 2</i>. Check the resources that perform this activity. List this activity if it is performed only by human resources</p> <p>Repeat for all activities identified in Step 2</p>		<p>This step identifies small activities performed by human resources, as activities executed by non-human resource and are impossible to redesign</p> <p>Result of the step: list of small activities performed by human resources</p>
	<p>Output: List of small activities performed by human resources</p>			
6. Redesign possibilities	<p>- Combine several small tasks into one composed activity</p>			

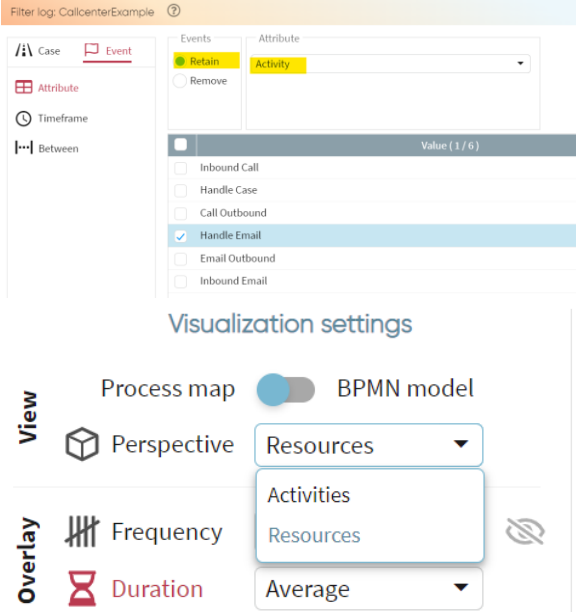
7. References	
7.1. References to the academic literature	[5]

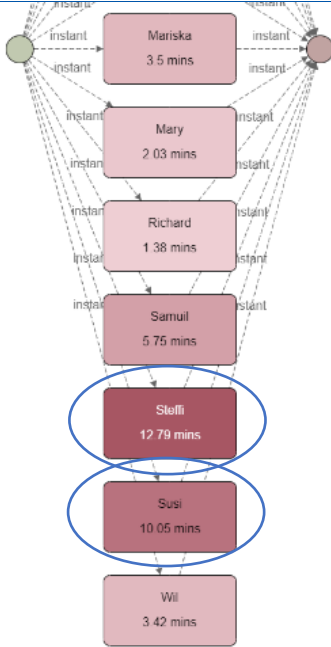
1. Improvement opportunity (IO)	Large activities		
2. Definition	An activity that has many functional procedures and long processing time		
3. Examples	In the insurance claims handling process: - Activity “Assess claim” is a large activity because it takes a long time to perform and has many functional procedures		
4. Minimum data needed	Activities, start timestamps, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allow determining the list of large activities		
	#	Step	<div> <div>Apromore example</div> <div> <div>Visualization settings</div> <div> <div>View</div> <div> <div>Process map</div> <div>BPMN model</div> </div> <div> <div>Perspective</div> <div>Activities</div> </div> </div> <div> <div>Overlay</div> <div> <div>Frequency</div> <div>Case</div> </div> <div> <div>Duration</div> <div>Average</div> </div> </div> </div> </div> <div>Explanation</div>
	1	1) Open event log in the process discoverer 2) In the <i>Visualization settings</i> , select <i>Duration overlay</i> and choose <i>Average</i> . In the <i>View</i> section, choose the <i>Activities</i> perspective	Result of the step: generated process map based on average duration and activities perspective
	2	From the process map, find activities that have the longest processing time and that can	 <p>The darker the activity is the longer is the processing time. On the example screenshot, large activity is circled in blue</p>

	potentially be decomposed into smaller tasks. List these activities		Result of the step: list of large activities
	Output: List of large activities		
6. Redesign possibilities	<ul style="list-style-type: none"> - Decompose large activity into several small ones - Divide a general activity into several alternative tasks that are better aligned with the capabilities of resources and types of cases 		
7. References 7.1. References to the academic literature	[5]		

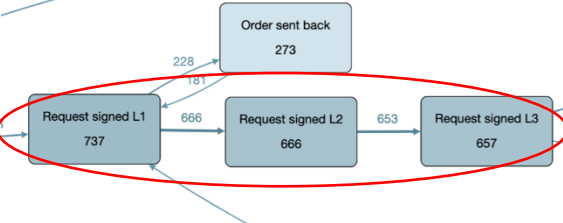
1. Improvement opportunity (IO)	Activity variants		
2. Definition	An activity that has abnormal variation in process time and/or waiting time based on the resource type or case attribute		
3. Examples	In the child delivery process: - There is a dependency between which doctor performs epidural delivery and the costs.		
4. Minimum data needed	Activities, start timestamps, end timestamps, resources, case attributes		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow determining if process contains activity variants, Step 4 provides overview of abnormal activity variants based on resources and/or case attributes		
	#	Step	Explanation
	1	Launch dashboard and navigate to Activities tab	 <p>Result of the step: generated dashboard</p>
	2	In the Activities statistics table (the last table on Activities tab), compare median and average duration of each activity. List activities that have a big difference between average and median durations	 <p>By comparing average and median durations of activities, it is possible to identify outliers that may have irregular variation in performance</p> <p>Result of the step: list of activities that have outliers in terms of processing time</p>
	3	1) Open event log in the process discoverer	 <p>Result of the step: list of activities with significant number of outliers (abnormal variation) in terms of processing and waiting time</p>

		<p>2) Open the <i>Filter log</i> screen. In Event → <i>Attribute</i> filter, select one activity from <i>Step 2</i>. Click OK.</p> <p>3) Add another filter: in Case → <i>Performance</i> filter, select <i>Average processing time</i> and then <i>Average waiting time</i> as Performance measure.</p> <p>4) Analyze both obtained distributions, if there are significant number of outliers in either or both options, list this activity</p> <p>Repeat for all activities identified in <i>Step 2</i></p>		
	4	<p>1) Open event log in the process discoverer</p>		<p>On the example screenshot, process map for selected activity is based on resources perspective and average duration. Resources <i>Steffi</i> and <i>Susi</i></p>

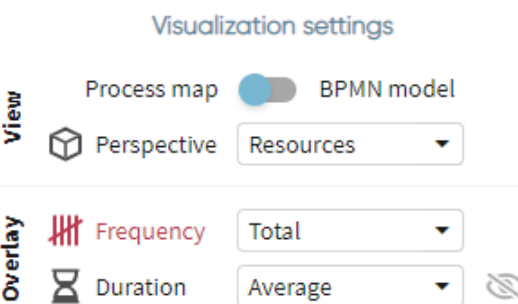
	<p>2) Open the <i>Filter log</i> screen. In Event → <i>Attribute</i> filter, select one activity from <i>Step 3</i>. Click OK.</p> <p>3) In the <i>Visualization</i> <i>settings</i>, select <i>Frequency</i> or <i>Duration overlay</i>. In the <i>View</i> section, choose the perspective (select those based on the process, case attributes available in event log, aim of the research, etc.).</p> <p>4) Find abnormal (extreme) cases in terms of resources and/or case attributes, and list them</p> <p>Repeat for all activities identified in <i>Step 3</i></p>		<p>are performing this activity longer than others</p> <p>Result of the step: list of abnormal activity variants based on resources and/or case attributes</p>
--	--	--	---

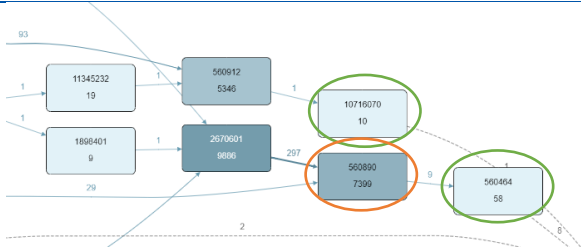
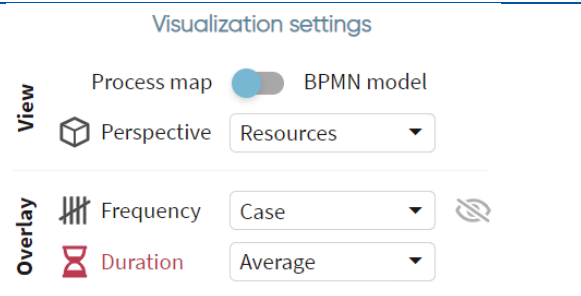
				
	Output: List of abnormal activity variants based on resources and/or case attributes			
6. Redesign possibilities	<ul style="list-style-type: none"> - Permanently set resources to execute a certain set of activities based on the highest efficiency, the lowest costs, the shortest performance time, etc. 			
7. References				
7.1. References to the academic literature	[6]			

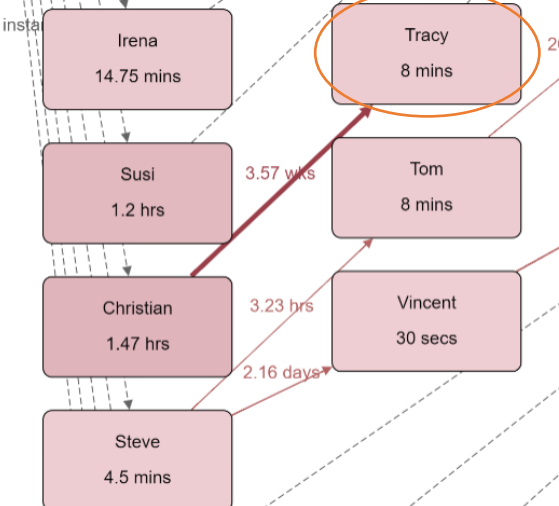
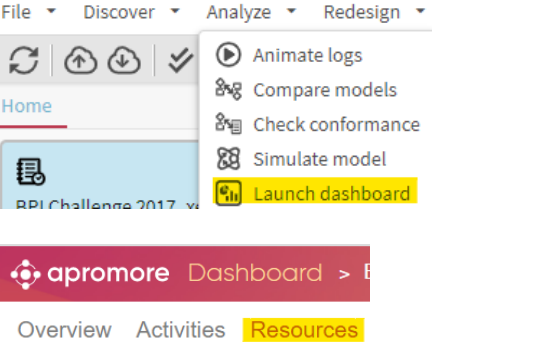
1. Improvement opportunity (IO)	Similar activities			
2. Definition	Two or more activities that have similar procedures			
3. Examples	In the car rental process: <ul style="list-style-type: none">- There are several check activities that are executed on different parts of the process, each of these activities requires set-up time and work handover. Check activities follow similar procedures, but they have different subject of check (parts of the car, lessee, etc.), require different data to perform the check, and these checks are performed by different resources (clerk, mechanic)			
4. Minimum data needed	Activities, end timestamps			
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allow determining the list of similar activities			
	#	Step	Apromore example	Explanation
	1	1) Open event log in the process discoverer 2) In the Visualization settings, select Frequency overlay and choose Case. In the View section, choose the Activities perspective	<div><div>Visualization settings</div><div><div>View</div><div><div>Process map <input checked="" type="checkbox"/> BPMN model</div><div><div>Perspective</div><div>Activities</div></div></div></div><div><div>Overlay</div><div><div>Frequency</div><div>Case</div></div><div><div>Duration</div><div>Average</div></div><div></div></div></div>	Result of the step: generated process map based on case frequency and activities perspective

	<p>2 From the process map, find activities that have similar procedures: look for activities with similar wording (e.g., check, verify). List sets of these activities</p>		<p>On the example screenshot, similar activities circled in red. They share similar procedure for signing request, but performed by different resources (L1, L2, L3).</p> <p>Result of the step: list of sets of similar activities</p>
<p>Output: List of sets of similar activities</p>			
<p>6. Redesign possibilities</p>	<ul style="list-style-type: none"> - Rearrange the process in a way that similar activities are placed close to each other, which will reduce setup time - Combine several similar tasks into one composed activity 		
<p>7. References</p> <p>7.1. References to the academic literature</p>	<p>[7, 8]</p>		

Resource-related improvement opportunities

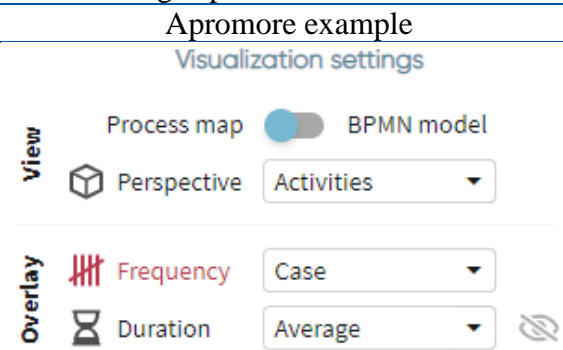
1. Improvement opportunity (IO)	High/low resource utilization		
2. Definition	High/low percentage of non-occupied resources in a process over the total number of resources		
3. Examples	In the payroll management process: <ul style="list-style-type: none"> - Too high workload of payroll specialists (documents compiling, SAP data registration, communication with clients), which they cannot handle 		
4. Minimum data needed	Activities, start timestamps, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: <i>Steps 1-4</i> allow determining underused and overused resources. There are two options provided that give the same output; the first is based on the process map analysis (<i>Steps 1-4</i>), the second is based on dashboard information analysis (<i>Steps 1-2</i>).		
	#	Step	Apromore example
		<i>Option 1</i>	
	1	1) Open event log in the process discoverer 2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Resources</i> perspective. Select <i>Frequency</i> overlay and choose <i>Total</i>	
			Result of the step: generated process map based on total frequency and resources perspective

	<p>2</p> <p>From the process map,</p> <ol style="list-style-type: none"> 1) Find resources with low case frequency (underused). List these resources 2) Find resources with high case frequency (overused). List these resources 		<p>Underused resource has low total frequency of cases (on the example screenshot, they are circled in green). Overused resource has high total frequency of cases (on the example screenshot, circled in orange)</p> <p>Result of the step: list of underused and overused resources</p>
	<p>3</p> <p>In the View section of Visualization settings, choose <i>Process map</i> and <i>Resources</i> perspective. Select <i>Duration overlay</i> and choose <i>Average</i></p>		<p>Result of the step: generated process map based on average duration and resources perspective</p>

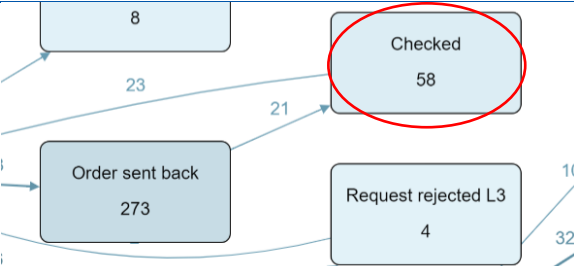
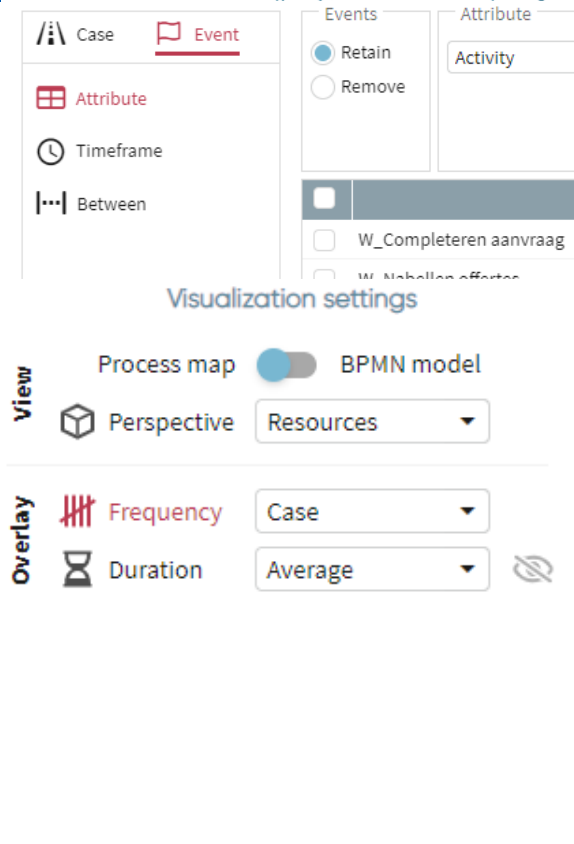
	<p>4 From the process map generated in <i>Step 3</i>, find resources with long waiting times on the incoming arcs (overused). List these resources if they were not listed in <i>Step 2</i></p>		<p>One more manifestation of overused resources is long waiting times on the incoming arcs (on the example screenshot, circled in orange; <i>Tracy</i> is an overused resource because it has long waiting time) The thicker the arrow, the longer is waiting time.</p> <p>Result of the step: complete list of overused resources</p>
	<p><i>Option 2 (does not account for overused resources based on long waiting times on incoming arcs)</i></p>		
	<p>1 Launch dashboard and navigate to Resources tab</p>		<p>Result of the step: generated dashboard</p>

	2	In the Resource statistics table (the last table on Resources tab), ascending-sort <i>Frequency</i> column to define the list of underused resources; then descending-sort <i>Frequency</i> column to define the list of overused resources	<table><tr><th colspan="4">Resource ▾</th></tr><tr><th>Value</th><th>Cases</th><th>Activity instances</th><th>▾ Frequency</th></tr><tr><td>560872</td><td>762</td><td>12117</td><td>23.21%</td></tr><tr><td>2670601</td><td>427</td><td>9886</td><td>18.93%</td></tr><tr><td>3273854</td><td>372</td><td>9075</td><td>17.38%</td></tr><tr><td>560890</td><td>611</td><td>7399</td><td>14.17%</td></tr><tr><td>560912</td><td>404</td><td>5346</td><td>10.24%</td></tr><tr><td>560925</td><td>206</td><td>1782</td><td>3.41%</td></tr><tr><td>11744364</td><td>72</td><td>1447</td><td>2.77%</td></tr><tr><td>560462</td><td>59</td><td>1443</td><td>2.76%</td></tr></table>	Resource ▾				Value	Cases	Activity instances	▾ Frequency	560872	762	12117	23.21%	2670601	427	9886	18.93%	3273854	372	9075	17.38%	560890	611	7399	14.17%	560912	404	5346	10.24%	560925	206	1782	3.41%	11744364	72	1447	2.77%	560462	59	1443	2.76%	Resources with high frequency are overused, resources with low frequency are underused Result of the step: list of underused and overused resources
Resource ▾																																												
Value	Cases	Activity instances	▾ Frequency																																									
560872	762	12117	23.21%																																									
2670601	427	9886	18.93%																																									
3273854	372	9075	17.38%																																									
560890	611	7399	14.17%																																									
560912	404	5346	10.24%																																									
560925	206	1782	3.41%																																									
11744364	72	1447	2.77%																																									
560462	59	1443	2.76%																																									
	Output: List of underused and overused resources																																											
6. Redesign possibilities	<ul style="list-style-type: none">- Decrease/increase the number of resources- Increase the specialization of human resources so that a person will be responsible only for a certain activities or cases- Tasks automatization- Allow customers to execute some parts of the process by themselves																																											
7. References																																												
7.1. References to the academic literature	[9, 10]																																											

Activity- and resource-related improvement opportunities

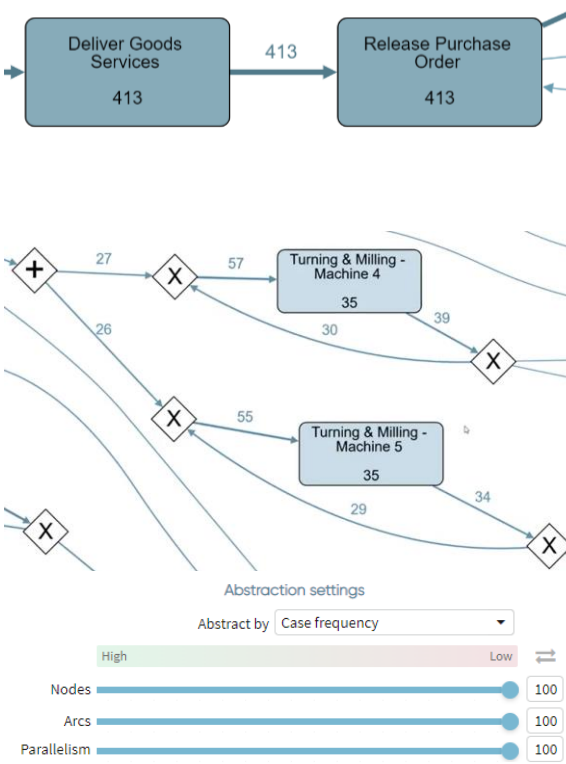
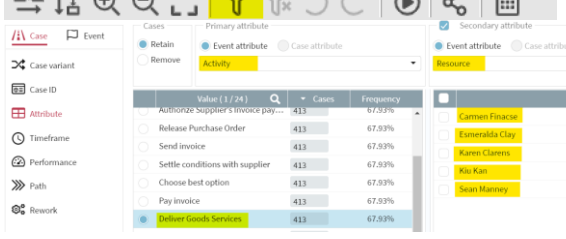
1. Improvement opportunity (IO)	Controls performed by internal resources		
2. Definition	An activity that has one or several functional procedures and short processing time		
3. Examples	In the billing process: <ul style="list-style-type: none"> - Billing checks are performed by internal controllers. If moved towards customers, it would eliminate company's errors. 		
4. Minimum data needed	Activities, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow determining if process contains controls that are executed by internal resources		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the <i>Visualization settings</i> , select <i>Frequency overlay</i> and choose <i>Case</i> . In the <i>View</i> section, choose the <i>Activities</i> perspective	

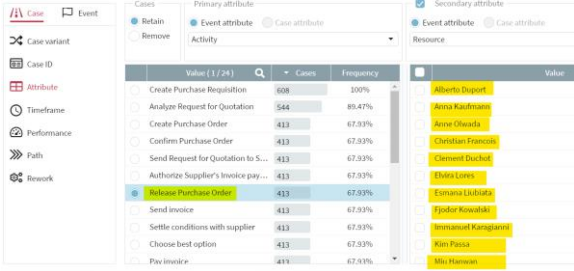
Result of the step: generated process map based on case frequency and activities perspective

	<p>2 From the process map, find check activities. List these activities</p>		<p>Look for activities that contain “check”, “verify”, or similar wordings.</p> <p>Result of the step: list of check activities</p>
	<p>3</p> <ol style="list-style-type: none"> 1) Open the <i>Filter log</i> screen. 2) Using <i>Event</i> → <i>Attribute</i> filter, select a check activity obtained in Step 2. Click OK. 3) In the <i>Visualization settings</i>, select <i>Frequency overlay</i> and choose <i>Case</i>. In the <i>View</i> section, choose the <i>Resources</i> perspective. 4) Check if activity is performed by internal resource(s). 5) List this activity if it is performed by 		<p>Result of the step: list of controls that are executed by internal resources</p>

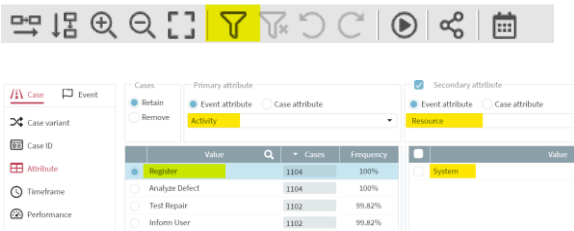
		internal resource(s).		
		Repeat for all identified in <i>Step 2</i> check activities.		
	Output: List of controls that are executed by internal resources			
6. Redesign possibilities	- Rearrange the process in a way that controls are performed by customers			
7. References				
7.1. References to the academic literature	[3]			

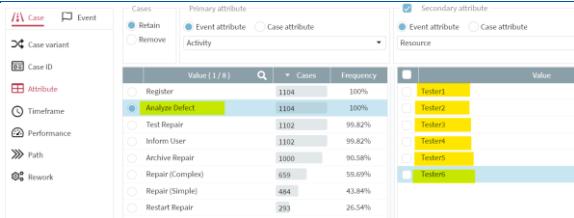
1. Improvement opportunity (IO)	Independent sequential activities		
2. Definition	Activities that are executed in a sequence while they are not dependent on each other in terms of inputs, outputs and, resources		
3. Examples	In the car rental process: <ul style="list-style-type: none">- Activities “Contract Preparation” and “Car Preparation” are executed in a sequence. These activities are independent in terms of input and output as well as resources (mechanic prepares a car, clerk prepares a contract)		
4. Minimum data needed	Activities, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow determining the list of independent sequential activities		
	#	Step	<div><div>Apromore example</div><div><div>Visualization settings</div><div><div>Process map</div><div><div>BPMN model</div></div></div><div><div>View</div><div><div>Perspective</div><div>Activities</div></div></div><div><div>Overlay</div><div><div>Frequency</div><div>Case</div></div><div><div>Duration</div><div>Average</div></div></div><div><div></div></div></div></div>
	1	<div>1) Open event log in the process discoverer</div> <div>2) In the Visualization settings, select BPMN model. Select Frequency overlay and choose Case. In the View section, choose the Activities perspective</div>	<div><div>Result of the step:</div> generated BPMN model based on case frequency and activities perspective</div>

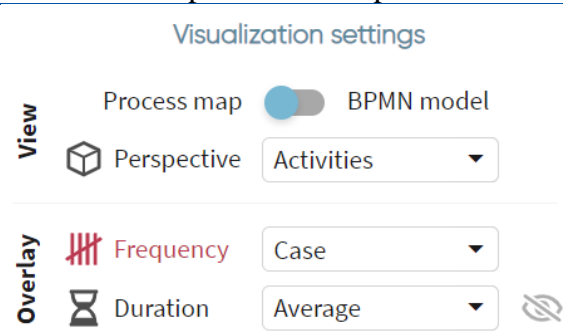
	<p>2</p> <p>From the BPMN model, define two or more sequential activities (activities are performed in parallel if they are executed in the parallel gateway (+ sign); these activities are not sequential) that do not depend on each other from the perspective of inputs and outputs (based on activity names). List activities of each fragment</p> <p>If needed, increase the percentage of arcs and parallelism in the Abstraction settings (see the second screenshot) to have a more detailed overview.</p>		<p>Analyze the process to define process fragments that are independent</p> <p>On the first example screenshot, activities <i>Deliver Goods Services</i> and <i>Release Purchase Order</i> are sequential and independent based on the activity names.</p> <p>On the second screenshot, activities <i>Turning & Milling – Machine 4</i> and <i>Turning & Milling – Machine 5</i> are NOT sequential because they are executed parallelly (parallel gateway)</p> <p>Result of the step: list of fragments with potential independent sequential activities</p>
	<p>3</p> <p>1) Open the <i>Filter log</i> screen.</p> <p>2) In Case → <i>Attribute</i> filter, set Primary attribute as <i>Event attribute, Activity</i>; set Secondary</p>		<p>Sequential activities are independent when they are executed by different resources (in addition to condition of having independent inputs and outputs). If all the activities of the fragment are done by different resources, they are independent</p>

	<p>attribute as <i>Event attribute</i>, <i>Resource</i>.</p> <p>3) Select the first activity of the fragment, list what resources are performing this activity; repeat for all activities of the fragment.</p> <p>4) List this fragment if all activities are executed by different resources</p> <p>Repeat for all process fragments listed in Step 2.</p>		<p>Result of the step: list of fragments with independent sequential activities</p>
<p>6. Redesign possibilities</p>	<p>Output: List of fragments with independent sequential activities</p> <p>- Rearrange the process in a way that independent sequential activities are executed in parallel</p>		
<p>7. References</p> <p>7.1. References to the academic literature</p>	<p>[11]</p>		

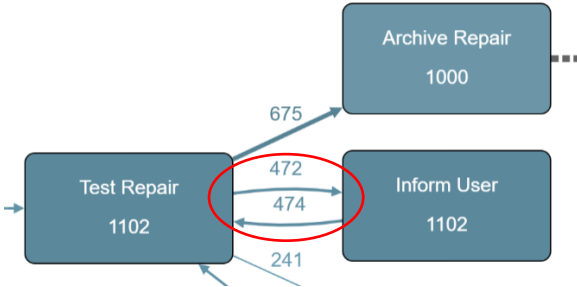
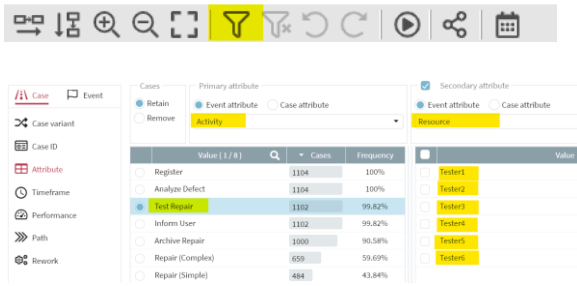
1. Improvement opportunity (IO)	Frequent handovers		
2. Definition	Transferring the case from one resource to another (as part of the business process or for the purpose of check, control, verification, etc.)		
3. Examples	<p>In the incident and problem management process:</p> <ul style="list-style-type: none"> - One support team is not able to solve the incident and reassigns it to another team or support line <p>In the pharmacy procurement process:</p> <ul style="list-style-type: none"> - An order is transferred from procurement to finance department to verify the supplier 		
4. Minimum data needed	Activities, resources, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow identifying frequent handovers and what activities correspond to them		
	#	Step	Apromore example
	1	<p>1) Open event log in the process discoverer</p> <p>2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Frequency</i> overlay and choose <i>Case</i></p>	<p>Visualization settings</p> <p>View Process map <input checked="" type="checkbox"/> BPMN model</p> <p>Perspective Activities</p> <p>Overlay Frequency Case</p> <p>Duration Average</p>
	2	From the process map, find arcs between the activities with the highest frequencies (the thicker	<p>Register 1104 → Analyze Defect 1104 → Repair (Complex) 659</p> <p>1104</p> <p>528</p>
		<p>Result of the step: generated process map based on case frequency and activities perspective</p> <p>To identify frequent handovers, firstly define activities between which cases are transferred more</p>	

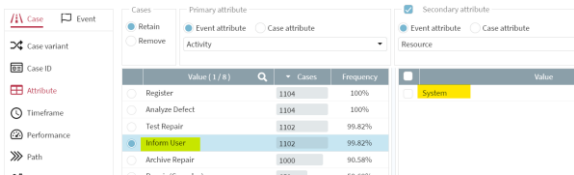
		the arrow, the higher the frequency). List activities pairs, arcs between which have the highest case frequencies		frequently (on the example screenshot, circled in red)
	3	<p>To define resources performing pairs of activities listed in <i>Step 2</i>, and check if there are handovers between them:</p> <ol style="list-style-type: none"> 1) Open the <i>Filter log</i> screen. 2) In Case → <i>Attribute filter</i>, set Primary attribute as <i>Event attribute</i>, Activity; set Secondary attribute as <i>Event attribute</i>, Resource 3) Select the first activity of the pair identified in <i>Step 2</i>, list resources that perform it 4) Select the second activity of the same pair identified in <i>Step</i> 		<p>Handovers are those parts of the process when a case circulates from one resource to another between activities. To identify handovers, check if activities are performed by different resources.</p> <p>In the example, pair of activities <i>Repair</i> → <i>Analyze Defect</i> is examined:</p> <ul style="list-style-type: none"> • <i>Repair</i> is performed by System • <i>Analyze Defect</i> is performed by six different testers <p>It can be concluded that there is a handover between System and Testers, which occurs between Repair and Analyze Defect activities.</p> <p>Result of the step: list of frequent handovers and between which pairs of activities they occurred</p>

	<p>2, list resources that perform it</p> <p>5) If activities from 3) and 4) are performed by different resources, list this pair of activities and resources that are performing them</p> <p>6) Click Cancel</p> <p>Repeat for all the pairs of activities identified in Step 2</p>		
	Output: list of frequent handovers and between which pairs of activities they occurred		
6. Redesign possibilities	<ul style="list-style-type: none"> - Minimize the number of resources involved in the process - Reorder the tasks in the process to minimize transfer handover - Reorder the process in a way to have one employee responsible for each type of case - Eliminate redundant tasks - Reduce middle management by providing employees the decision-making authority 		
7. References			
7.1. References to BPIC submissions	[12, 13, 14]		
7.2. References to the academic literature	[15, 16]		

1. Improvement opportunity (IO)	Ping pong behavior		
2. Definition	Transferring the case between two consecutive activities from one resource to another		
3. Examples	In the incident and problem management process: <ul style="list-style-type: none"> - The first support team is not able to solve the incident and reassigns it the second another team, which also struggles to fix the issue and assigns it back to the first support team 		
4. Minimum data needed	Activities, resources, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow identifying ping pongs from activities perspective and defining list of resources that are performing each activity		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Frequency</i> overlay and choose <i>Case</i>	

Result of the step: generated process map based on case frequency and activities perspective


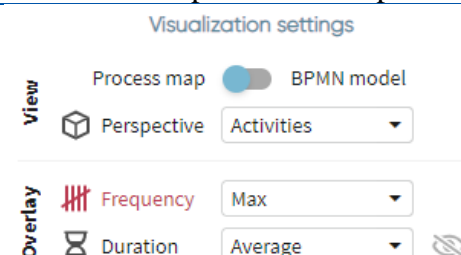
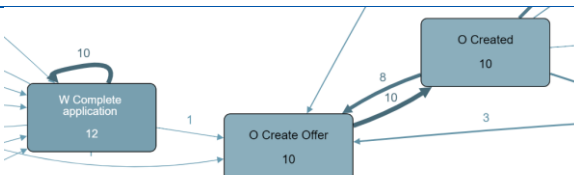
	<p>2 From the process map, find pairs of consecutive activities between which cases are circulated back and forth. List these pairs of activities</p>		<p>Ping pongs are those parts of the process when a case circulates from one resource to another back and forth, from one activity to another. This step identifies ping pongs from activities perspective (on the example screenshot, potential ping pong is circled in red)</p> <p>Result of the step: list of pairs of activities with potential ping pong behavior</p>
	<p>3 To define resources performing pairs of activities listed in <i>Step 2</i>, and check if there is a ping pong effect between them:</p> <ol style="list-style-type: none"> 1) Open the <i>Filter log</i> screen. 2) In Case → <i>Attribute filter</i>, set Primary attribute as <i>Event attribute, Activity</i>; set Secondary attribute as <i>Event attribute, Resource</i> 3) Select the first activity of the pair identified in <i>Step</i> 		<p>Mandatory condition for ping pong behavior is transferring case between different resources. This step checks if this condition holds by identification of the list of resources that performed each activity from <i>Step 2</i>.</p> <p>In the example, pair of activities <i>Test Repair</i> → <i>Inform User</i> is examined:</p> <ul style="list-style-type: none"> • <i>Test Repair</i> is performed by six different testers • <i>Inform User</i> is performed by System <p>It can be concluded that there is a ping pong between System and Testers, which occurs between Test Repair and Inform User activities.</p>

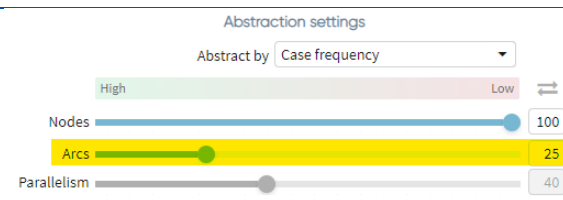
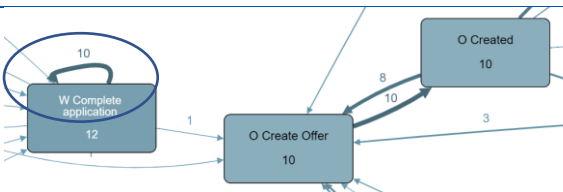
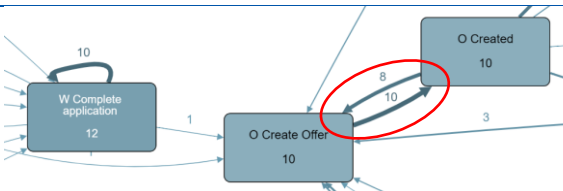
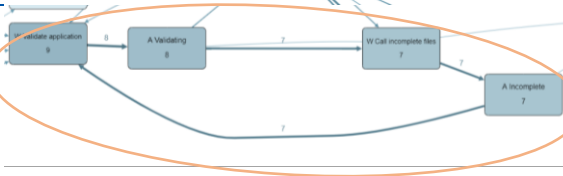
	<p>2, list resources that perform it</p> <p>4) Select the second activity of the same pair identified in <i>Step 2</i>, list resources that perform it</p> <p>5) If activities from 3) and 4) are performed by different resources, list this pair of activities and resources that are performing them</p> <p>6) Click Cancel</p> <p>Repeat for all the pairs of activities identified in Step 2</p>		<p>Result of the step: list of resources with ping pong behavior and between which pairs of activities it occurred</p>
	Output: list of pairs of resources with ping pong behavior and list of activities these resources are performing		
6. Redesign possibilities	<ul style="list-style-type: none"> - Minimize the number of resources involved in the process - Reorder the process in a way to have one employee responsible for each type of case - Eliminate redundant tasks - Reduce middle management by providing employees the decision-making authority 		
7. References 7.1. References to BPIC submissions	[12, 13, 14]		

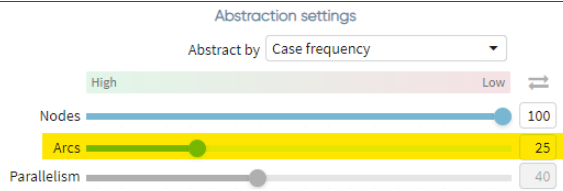
7.2. References to the academic literature

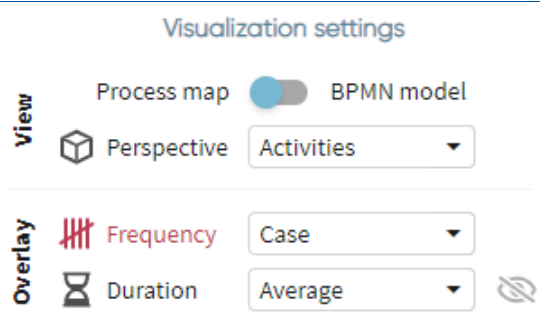
[15, 16]

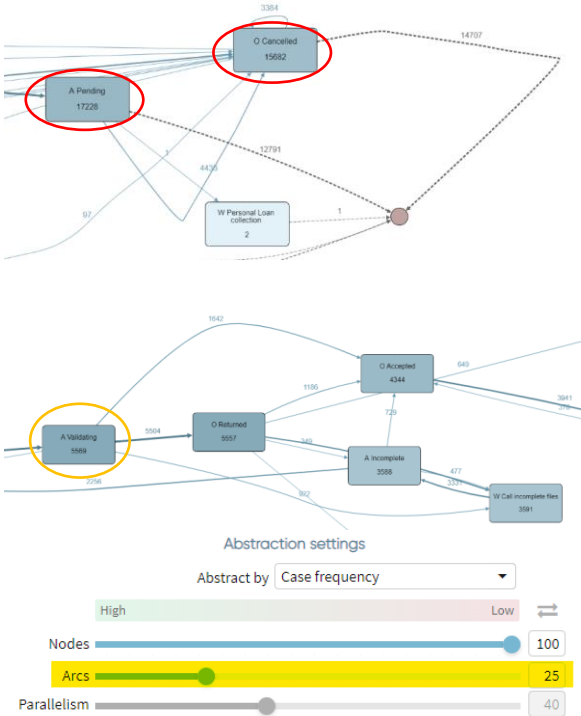
Control flow-related improvement opportunities

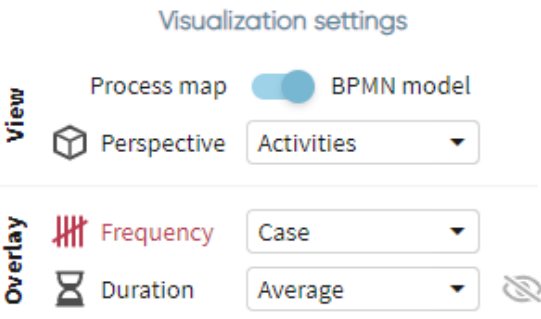
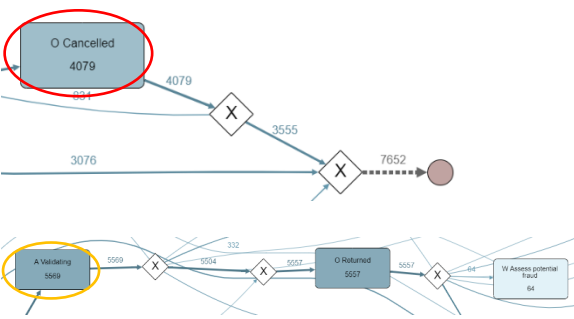
1. Improvement opportunity (IO)	Rework			
2. Definition	An activity or a fragment of the process that is repeated within one case			
3. Examples	In the order handling process: <ul style="list-style-type: none">- Renegotiating purchase orders that were initially unfeasible but the fact that requirements cannot be met is detected only in the phase of reviewing by purchasing department			
4. Minimum data needed	Activities, end timestamps			
5. Guideline on how to identify this IO	Expected output: Reworks based on activities and fragments are found using Steps 1-3.			
	#	Step	<div>Visualization settings</div> <div><div>View</div><div><div>Process map <input checked="" type="checkbox"/> BPMN model</div><div><div>Perspective</div><div>Activities</div></div></div></div> <div><div>Overlay</div><div><div>Frequency</div><div>Max</div></div><div><div>Duration</div><div>Average</div></div></div> <div></div>	Explanation
	1	<div>1) Open event log in the process discoverer</div> <div>2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Frequency</i> overlay and choose <i>Max</i></div>		<div>Rework is identified based on the max frequency the activity was repeated</div> <div>Result of the step: generated process map based on maximum frequency and activities perspective</div>
2	From the process map, find activities with frequency of execution higher than 1.		Result of the step: list of reworks based on activities	

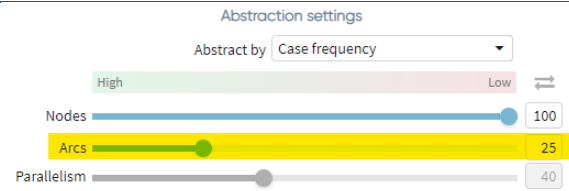
		<p>If needed, increase the percentage of arcs in the Abstraction settings (see the second screenshot) to have a more detailed overview.</p>		
3	<p>From the process map, for the activities with frequency >1 identified in <i>Step 2</i>,</p> <p>1) Find activities with self-loop arcs. List these activities as self-loop reworks.</p>		<p>Self loop occurs when one activity is repeated several times (see the first example circled in blue).</p>	
	<p>2) Find activities with short loops. List these activity pairs as short-loop reworks.</p>		<p>Short loop (ping pong) occurs when two activities are repeated several times (see the first example circled in red).</p>	
	<p>3) Find fragments with indirect repetition loops that have arcs frequency more than 1. List activities of these fragments as fragment reworks.</p>		<p>Indirect repetition loop occurs when several activities are repeated in a sequence (see the second example circled in orange)</p> <p>Result of the step: list of self-loop, short-loop and fragment reworks</p>	

	<p>If needed, increase the percentage of arcs in the Abstraction settings (see the last screenshot) to have a more detailed overview.</p>		
	<p>Output: list of reworks based on activities and fragments</p>		
6. Redesign possibilities	<ul style="list-style-type: none"> - Add control tasks to check if the case is executed correctly and with required quality 		
7. References			
7.1. References to BPIC submissions	[17, 18, 19]		
7.2. References to the academic literature	[20]		

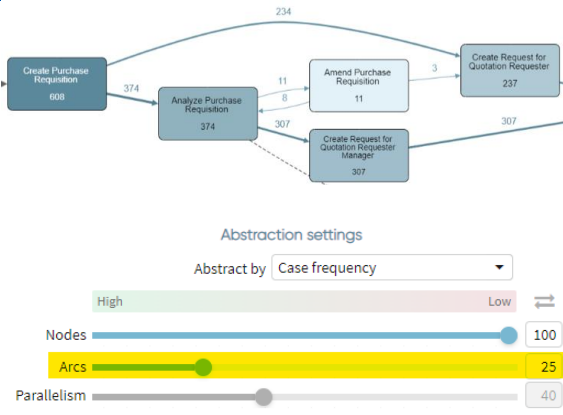
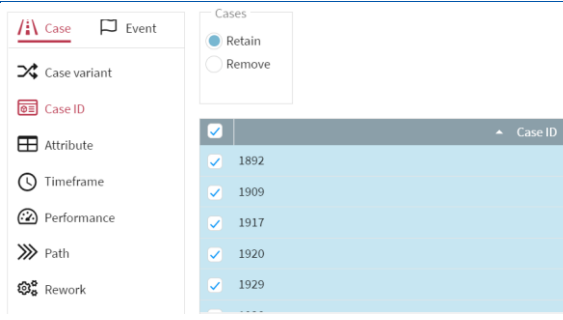
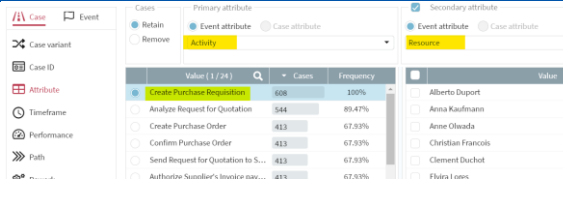
1. Improvement opportunity (IO)	Knock-out		
2. Definition	An activity that checks if the case should move forward with the process or if it should be rejected		
3. Examples	<p>In the loan application process:</p> <ul style="list-style-type: none"> - After the client submits a loan application, the clerk performs three checks in chaotic order: identity check, verification, and creditworthiness check. If any of these checks fail, the application is rejected. As there is no defined order in which checks should be done, the overprocessing issue arises due to wrongly placed knock-outs 		
4. Minimum data needed	Activities, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allow determining the list of knock-outs. There are two options provided on how to find knock-outs, both give the same output; the first option is based on process map analysis, the second is based on BPMN model analysis		
	#	Step	Apromore example
		<i>Option 1</i>	
	1	<p>1) Open event log in the process discoverer</p> <p>2) In the <i>Visualization settings</i>, select <i>Frequency overlay</i> and choose <i>Case</i>. In the <i>View</i> section, choose the <i>Activities</i> perspective</p>	 <p>Result of the step: generated process map based on case frequency and activities perspective</p>

	<p>2</p> <p>From the process map,</p> <ol style="list-style-type: none"> 1) Find activities with outgoing dashed arc to the end event that is the final activity of the process. List these activities 2) Find activities with outgoing arc to activity that is not the final activity of the process that leads to skipping one or several activities. List these activities <p>If needed, increase percentage of arcs to have a more detailed process map</p> <p><i>Option 2</i></p>		<p>There are two types of knock-outs:</p> <ol style="list-style-type: none"> 1) Knock-outs that lead to the end of the process (on the example screenshot, they are circled in red) 2) Knock-outs that lead to the later steps of the process, so that a part of the process is skipped (on the example screenshot, it is circled in orange; in 992 cases after <i>A Validating</i>, <i>W Call incomplete files</i> executes by skipping <i>O Returned</i> and <i>A Incomplete</i> activities) <p>Result of the step: list of knock-outs</p>
--	--	--	---

	1	<ol style="list-style-type: none"> 1) Open event log in the process discoverer 2) In the <i>Visualization settings</i>, select BPMN model. Select <i>Frequency overlay</i> and choose <i>Case</i>. In the <i>View</i> section, choose the <i>Activities</i> perspective 		<p>Result of the step: generated BPMN model based on case frequency and activities perspective</p>
	2	<p>From the process map,</p> <ol style="list-style-type: none"> 1) Find activities which are followed by two consecutive XOR splits and the end event. List these activities 2) Find activities which are followed by two consecutive XOR splits but do not lead to the process end event. List these activities <p>If needed, increase percentage of arcs to have</p>		<p>There are two types of knock-outs:</p> <ol style="list-style-type: none"> 1) Knock-outs that lead to the end of the process (on the example screenshot, circled in red) 2) Knock-outs that lead to the later steps of the process, so that a part of the process is skipped (on the example screenshot, it is circled in orange) <p>Result of the step: list of knock-outs</p>

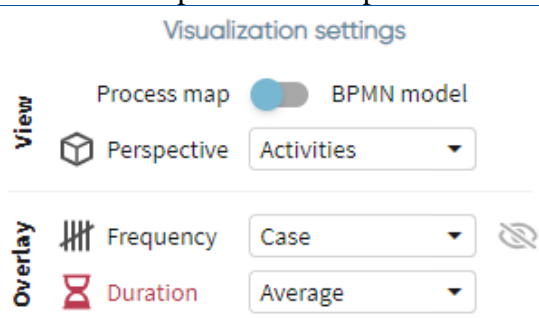
	a more detailed process map		
	Output: List of knock-outs		
6. Redesign possibilities	<ul style="list-style-type: none"> - Rearrange knock-outs in a way that, in the first place, will be knock-outs with the lowest effort needed and the highest termination probability - Rearrange the process in a way that knock-out checks will be early in the process - Reduce middle management by providing employees the decision-making authority 		
7. References			
7.1. References to the academic literature	[21]		

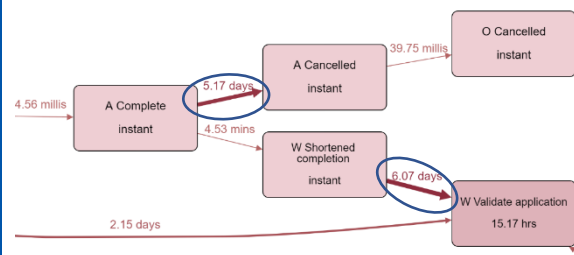
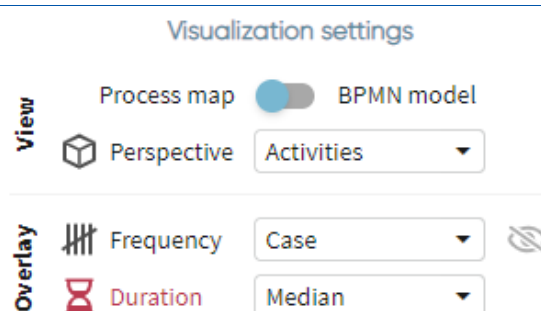
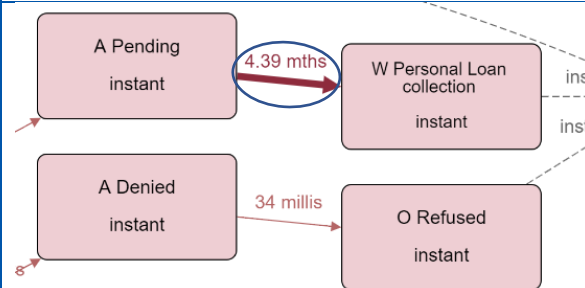
1. Improvement opportunity (IO)	Workaround		
2. Definition	Temporary process deviation from the standardized business process		
3. Examples	<p>In the course registration university process:</p> <ul style="list-style-type: none"> - The system does not support changing study programs after the semester starts. As a workaround, students who want to switch the program should turn to a secretary who will manually move course credits from one program to another one by one. This is time-consuming and leads to errors. 		
4. Minimum data needed	Activities, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: <i>Step 2</i> allows identifying bypass of process parts workarounds, <i>Step 3</i> – dummy instances workarounds, <i>Step 4</i> – in compliance to the position workarounds		
	#	Step	<div> <div> <div>Visualization settings</div> <div> <div> <div>Process map</div> <div><input checked="" type="checkbox"/></div> <div>BPMN model</div> </div> <div> <div>View</div> <div>Perspective</div> <div>Activities</div> </div> </div> <div> <div>Overlay</div> <div>Frequency</div> <div>Case</div> </div> <div> <div>Duration</div> <div>Average</div> <div></div> </div> </div> </div>
	1	1) Open event log in the process discoverer 2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Frequency</i> overlay and choose <i>Case</i> .	Result of the step: generated process map based on case frequency and activities perspective

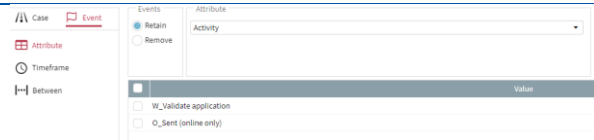
	<p>2 From the process map, find process fragments where one or several activities are skipped. List fragments with the highest case frequency on arcs that skip activities</p> <p>If needed, increase the percentage of arcs in the Abstraction settings (see the screenshot) to have a more detailed overview.</p>		<p>According to Outmazgin, & Soffer (2016) one of the typical types of workarounds is Bypass of process parts: one or several activities are bypassed by a process instance.</p> <p>On the example screenshot, in 234 cases after executing <i>Create Purchase Requisition</i>, <i>Create Request for Quotation Requester</i> is performed by skipping several other activities</p> <p>Result of the step: list of workarounds with cases that bypass process parts</p>
	<p>3 Open the <i>Filter log</i> screen. In Case → <i>Case ID</i> filter, check names of Case IDs. List Case IDs that are not following naming convention (e.g., 99999, 0000, test)</p>		<p>Another type of workarounds based on classification of Outmazgin, & Soffer (2016) is Dummy instances: creating fictitious instances for testing or monitoring purposes (e.g., 9999)</p> <p>Result of the step: list of dummy instances workarounds</p>
	<p>4 If it is known what resources are responsible for what activities or it is understandable from resource names: Open the <i>Filter log</i> screen.</p>		<p>Incompliance to the position is one more type of workarounds, it manifests when resources perform activities that are not under their responsibility (Outmazgin, & Soffer, 2016)</p>

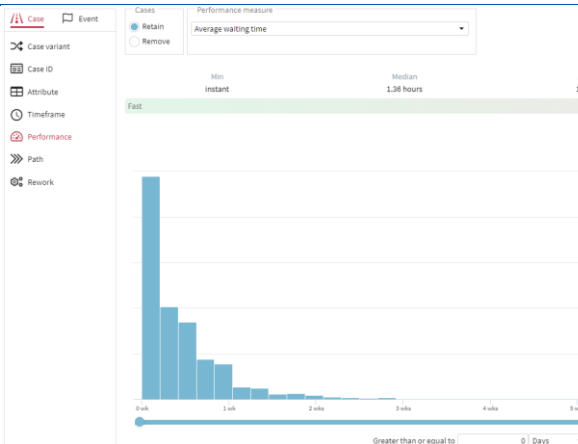
	<p>In Case → <i>Attribute</i> filter, set Primary attribute as <i>Event attribute, Activity</i>; set Secondary attribute as <i>Event attribute, Resource</i>. Select activities one by one and check the list of resources that perform its activities. List activities that are performed by resources that are not responsible for it</p>	<p>Result of the step: list activities that are performed by incorrect resources</p>
	Output: Lists of workarounds of different types	
6. Redesign possibilities	<ul style="list-style-type: none"> - Add missing system functionalities 	
7. References		
7.1. References to BPIC submissions	[22, 23, 24]	
7.2. References to the academic literature	[25, 26]	

Waiting time-related improvement opportunities

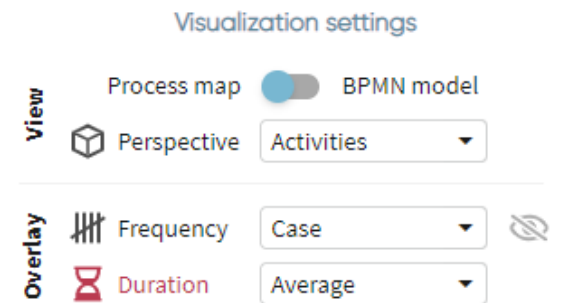
1. Improvement opportunity (IO)	Highest waiting time in the business process		
2. Definition	The longest average time that cases spend in an idle mode (waiting for further processing) between two activities		
3. Examples	<p>In the loan application process:</p> <ul style="list-style-type: none"> - Time between submitting a loan request and acceptance of this request (client waits until bank employee considers the request) - Time between sending an offer and checking the documents provided (bank waits for client to provide documents) - Time between loan request approval and receiving an offer (client waits for bank employee to send an offer) <p>In the healthcare process:</p> <ul style="list-style-type: none"> - Time between a nurse makes an inquiry about the health of the patient and the patient is assigned to the doctor 		
4. Minimum data needed	Activities, start timestamps, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-5 allow identifying the highest waiting times between activities in the process based on average and median waiting times and defining which of them is more robust. If a more detailed look is needed, Step 6 gives the distribution of waiting time for each case.		
	#	Step	<div> <div> <div>Apromore example</div> <div>  </div> </div> <div> <div>Explanation</div> <p>One of the ways to find the highest waiting time between the activities is by calculating the average duration between activities</p> <p>Result of the step: generated process map based on average duration</p> </div> </div>

		Select <i>Duration</i> overlay and choose <i>Average</i> .		
2	Analyze process map to find extremely high waiting times based on average duration. Find the thickest arrows between the activities. List the pairs of activities between which the thickest arrows are identified.		<p>The thicker the arrow between activities, the higher is waiting time. Identify the extreme cases (on the example screenshot, they are circled in blue)</p> <p>Result of the step: one or several highest waiting times based on average duration</p>	
3	In the Visualization settings, switch duration to <i>Median</i> , keep the rest of the settings the same as in Step 1.		<p>It is also possible to find the highest waiting time between the activities by calculating the median duration between activities</p> <p>Result of the step: generated process map based on median duration</p>	
4	Analyze process map generated in Step 3 to find extremely high waiting times based on median duration. Find the thickest arrows between the activities. List the pairs of activities between which the		<p>The thicker the arrow between activities, the higher is waiting time. Identify the extreme cases (on the example screenshot, they are circled in blue)</p> <p>Result of the step: one or several highest waiting times based on median duration</p>	

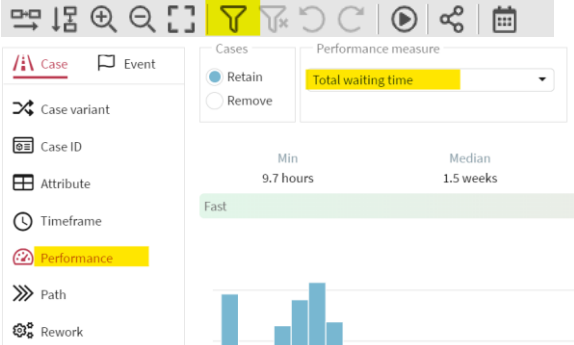
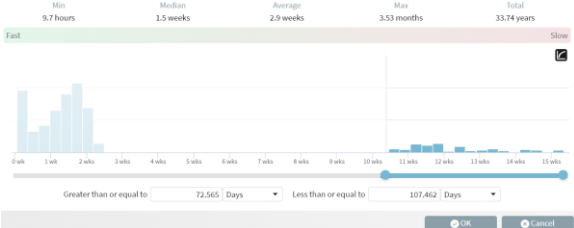
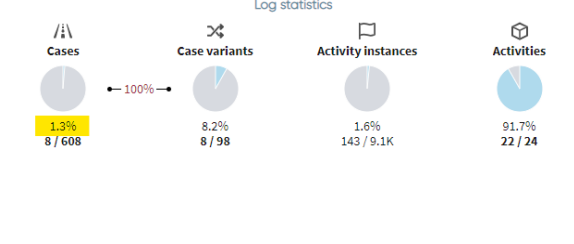
		thickest arrows are identified.		
	5	<p>Compare the results of Step 2 and Step 4. List all pairs of activities between which the highest waiting times from Step 2 and Step 4 identified. If duplication occurs, choose average or median value based on the rule following rule:</p> <ul style="list-style-type: none"> • if the average waiting time is more than 1.5-2 times higher than the median waiting time, select median one • if average waiting time is less than 1.5-2 times higher than the median waiting time, select average one 		<p>If the average waiting time is considerably higher than the median, it signs that the data has outliers which significantly influence the mean values. In this case median is a more robust parameter to define the longest waiting times</p> <p>Result of the step: final list of the highest waiting times</p>
	6	<p>(Optional step) Open the <i>Filter log</i> screen. Using <i>Attribute</i> filter, select a pair of activities from the list obtained in Step 5. Then, using</p>		<p>When waiting time is identified, to drill-down and analyze the case distribution based on waiting time, firstly, limit the log to investigated activities, then check the distribution of cases based on waiting time. In the</p>

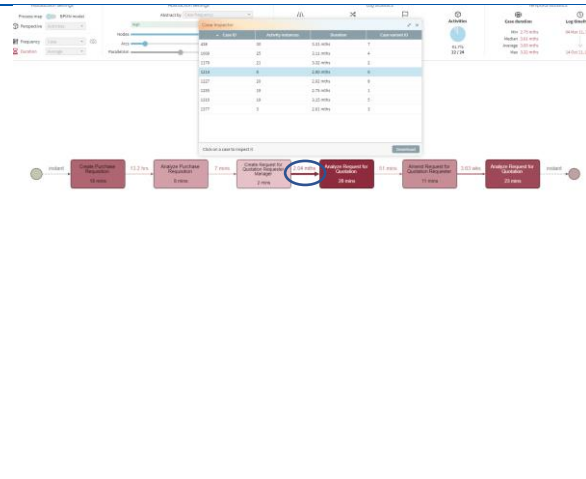
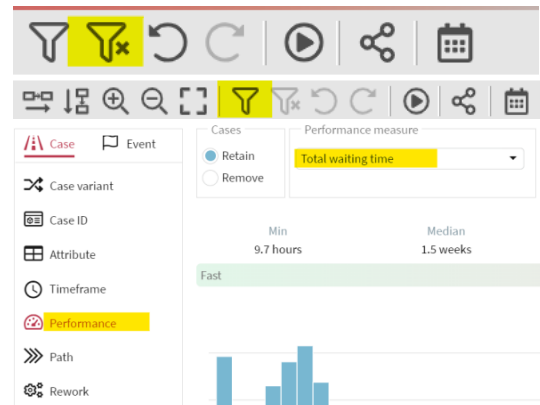
	<p><i>Performance filter, retain cases with Average or Median waiting time (based on the Step 5 result; if in Step 5 waiting time was taken based on average value, choose Average waiting time, if based on median value – choose Median waiting time).</i></p> <p>Identify the distribution of waiting time duration per number of cases; the more cases have a high duration, the more severe is the need for process improvement.</p> <p>Repeat for all pairs of identified activities with high waiting time between them.</p>		<p>presented example, most of cases have waiting time from instant to 1.5 days</p> <p>Result of the step: distribution of cases in terms of average/median waiting time for the identified cases in Step 5</p>
<p>Output: Identified the highest waiting times between activities</p>			
<p>6. Redesign possibilities</p>	<ul style="list-style-type: none"> - Change arrival rate pattern, e.g., by incentivizing customers to come in hours with light-load instead of peak hours - Change batching schedule by making batching more frequent - Change task composition by combining several small tasks into one - Start using a technological solution to reduce waiting time, e.g., workflow management system - Empower employees to make decisions and reduce middle management - Implement follow-ups and assistance to external customers 		

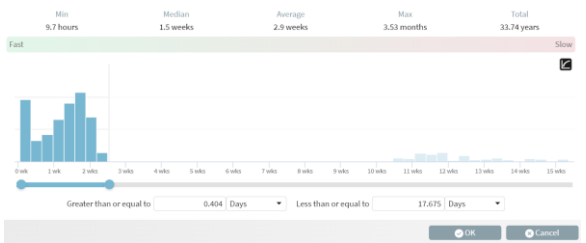
7. References	
7.1. References to BPIC submissions	[27, 28, 29]
7.2. References to the academic literature	[30, 31]

1. Improvement opportunity (IO)	Cases with the highest waiting times		
2. Definition	Cases that have the highest total time in an idle mode (waiting for further processing)		
3. Examples	<p>In the loan application process:</p> <ul style="list-style-type: none"> - Time between submitting a loan request and acceptance of this request (client waits until bank employee considers the request) - Time between sending an offer and checking the documents provided (bank waits for client to provide documents) - Time between loan request approval and receiving an offer (client waits for bank employee to send an offer) <p>In the healthcare process:</p> <ul style="list-style-type: none"> - Time between a nurse makes an inquiry about the health of the patient and the patient is assigned to the doctor 		
4. Minimum data needed	Activities, start timestamps, end timestamps		
5. Guideline on how to identify this IO	Expected output: <i>Steps 1-4</i> allow identifying case IDs with the highest overall waiting time. If a more detailed look is needed, <i>Step 5</i> gives the highest waiting time between activities for each case ID separately.		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Duration</i> overlay and choose <i>Average</i> .	

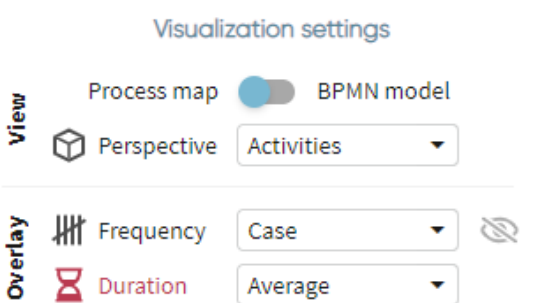
Templates for visual process analysis using process mining tools

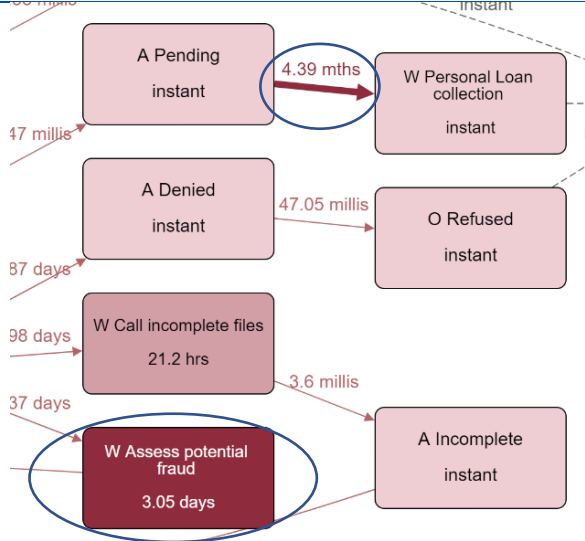
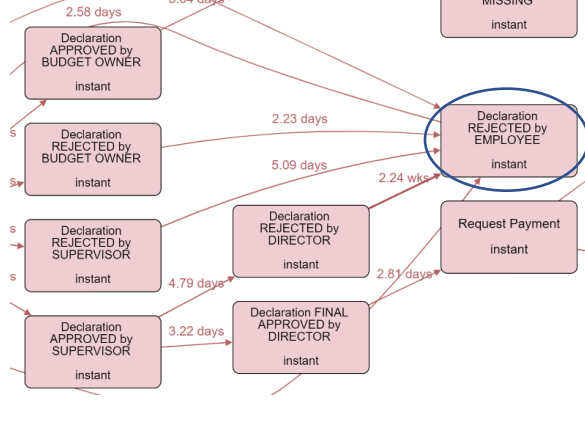
	<p>2 Open the <i>Filter log</i> screen. Choose Case → <i>Performance</i> filter. Select “Total waiting time” in Performance measure and “Retain” in Cases</p>		<p>Result of the step: visual distribution of cases based on total waiting time</p>
	<p>3 Explore the distribution from <i>Step 2</i> and slice it to the cases with the highest total waiting time. Click OK</p>		<p>Result of the step: generated process map based on cases with the highest waiting time</p>
	<p>4 Open <i>Case Inspector</i> by clicking on in Log statistics section List case IDs with the highest waiting time (those that were selected in <i>Step 3</i>)</p>		<p>Result of the step: list of case IDs highest waiting times</p>
	<p>5 (<i>Optional step</i>) To define the highest waiting times between activities for each of the cases: 1) Click on the case ID in <i>Case</i></p>		<p>The thicker the arrow between activities, the higher is waiting time.</p> <p>Result of the step: list of the highest waiting times between activities for each of the cases from <i>Step 4</i></p>

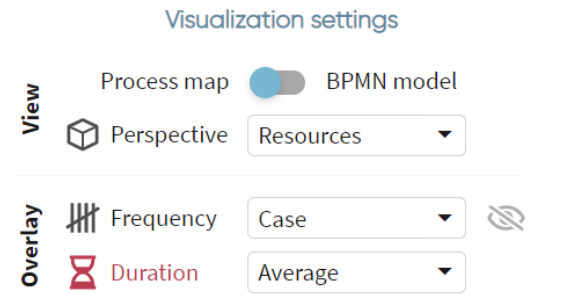
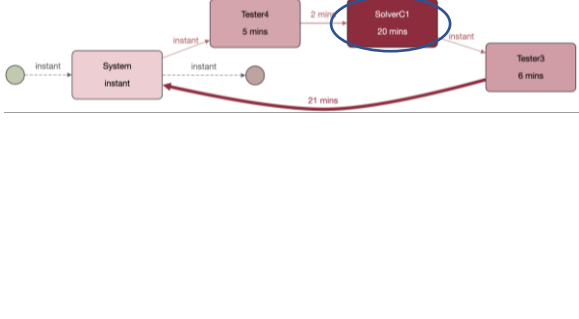
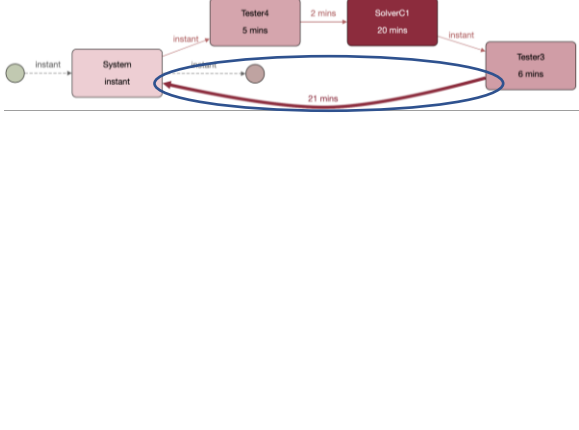
	<p><i>Inspector</i> from <i>Step 4</i></p> <p>2) Explore the generated process map and list the highest waiting time between activities for the chosen case</p> <p>Repeat for all of case IDs with the highest waiting time</p>		
6	<p>(Optional step)</p> <p>To identify differences between the process with cases with the highest waiting times and process with small waiting times:</p> <ol style="list-style-type: none">1) Clear all filter criteria2) Open the <i>Filter log</i> screen3) Choose Case → <i>Performance</i> filter. Select “Total waiting time” in Performance measure and “Retain” in Cases		<p>Result of the step: list of the notable differences between process map with cases with the highest total waiting time and process map with cases with low and moderate total waiting time</p>

	<ol style="list-style-type: none"> 4) Explore the distribution from slice it to the cases with low and moderate total waiting time. Click OK 5) Compare side-by-side generated process map with the one generated in <i>Step 3</i> 6) List all the notable differences (e.g., self-loops, short loops, indirect repetition loops with high waiting times; additional activities that trigger waiting) 		
6. Redesign possibilities		<p>Output: Identified cases the highest overall waiting time and the highest waiting time between activities for each case</p> <ul style="list-style-type: none"> - Change arrival rate pattern, e.g., by incentivizing customers to come in hours with light-load instead of peak hours - Change batching schedule by making batching more frequent - Change task composition by combining several small tasks into one - Start using a technological solution to reduce waiting time, e.g., workflow management system - Empower employees to make decisions and reduce middle management - Implement follow-ups and assistance to external customers 	

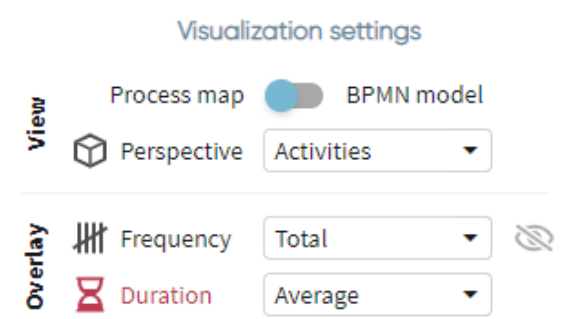

7. References	
7.1. References to BPIC submissions	[27, 28, 29]
7.2. References to the academic literature	[30, 31]

1. Improvement opportunity (IO)	Bottleneck		
2. Definition	A situation when the number of cases arriving exceeds the number of cases that can be handled which leads to queues or case build-ups		
3. Examples	<p>In the healthcare process:</p> <ul style="list-style-type: none"> - In the hospital, patients need to wait to make the computed tomography scan due to lack of personnel which causes queues - In the pharmacy, customers need to wait in the queue to get served because pharmacists are busy with serving other customers 		
4. Minimum data needed	Activities, resources, start timestamps, end timestamps		
5. Guideline on how to identify this IO	Expected output: Activity bottlenecks are found using <i>Steps 1-2</i> , resource bottlenecks are found using <i>Steps 3-4</i> .		
	#	Step	Apromore example
	1	<p>1) Open event log in the process discoverer</p> <p>2) In the <i>Visualization settings</i>, select <i>Duration overlay</i> and choose <i>Average</i>. In the <i>View</i> section, choose the <i>Activities</i> perspective</p>	 <p>The bottleneck is identified based on the long time needed to process the case or the long waiting time between activities</p> <p>Result of the step: generated process map based on average duration and activities perspective</p>

2	<p>From the process map,</p> <ol style="list-style-type: none"> 1) Find activity bottlenecks: activities with the longest processing time. List these activities. 2) Find waiting time bottlenecks: arcs with the longest duration. List activity pairs. 		<p>To find activity bottleneck, define activities with the longest duration. To find waiting time bottlenecks, find the longest waiting times (the thicker the arrow between activities, the higher is waiting time). On the example screenshot, they are circled in blue.</p>
	<ol style="list-style-type: none"> 3) Find resource-capacity bottlenecks: activities with the highest number of incoming arcs with long arc duration. List these activities. 		<p>To find resource-capacity bottlenecks, define activities all/most incoming arcs of which have long waiting time. On the example screenshot, it is circled in blue.</p> <p>Result of the step: list of activity bottlenecks, waiting time bottlenecks, and resource-capacity bottlenecks</p>

	<p>3 In the <i>Visualization settings</i>, select <i>Duration overlay</i> and choose <i>Average</i>. In the <i>View</i> section, choose the <i>Resources</i> perspective</p>		<p>Result of the step: generated process map based on average duration and resources perspective</p>
	<p>4 From the process map, 1) Find resource bottlenecks: resources with the highest processing time from the resource perspective. List these resources</p>		<p>To find resource bottleneck, define resources with the longest duration. On the example screenshot, it is circled in blue.</p>
	<p>2) Find waiting time bottlenecks from the resource perspective: arcs with the longest duration. List these pairs of resources</p>		<p>To find waiting time bottlenecks from the resource perspective, find the longest waiting times (the thicker the arrow between resources, the higher is waiting time). On the example screenshot, it is circled in blue.</p> <p>Result of the step: list of resource bottlenecks, and waiting time bottlenecks from the resource perspective</p>
<p>Output: List of bottlenecks based on activities and resources</p>			

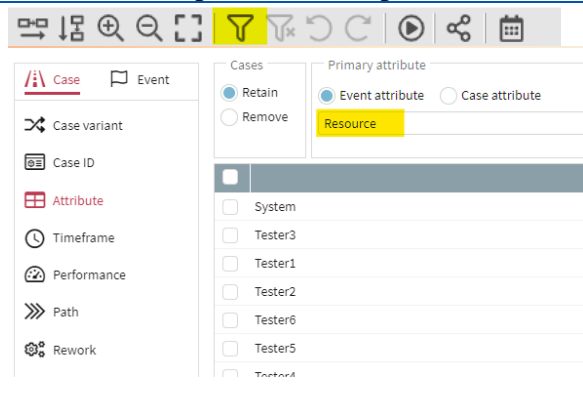
6. Redesign possibilities	<ul style="list-style-type: none">- Implement a technological solution to minimize constraints in the process- Organize separate process paths for different types of orders- Add more resources to the business process, if bottlenecks are caused by a lack of resources- Implement a scheduling system for clients to evenly distribute the workload, if bottlenecks are caused by queues during peak hours- Implement a resource scheduling system that will allow having more resources during peak hours and fewer resources during periods with low demand- Introduce a buffer queue- Use incentives to shift customers from high-demand hours to low-demand hours- Allow customers to execute some parts of the process by themselves
7. References	
7.1. References to BPIC submissions	[32, 33, 34]
7.2. References to the academic literature	[35, 36]


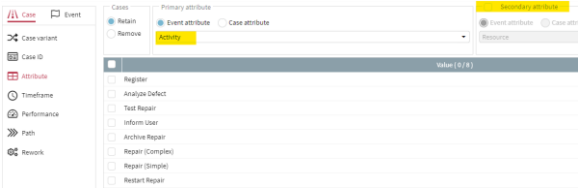
1. Improvement opportunity (IO)	Manual time-consuming fragment		
2. Definition	Business process fragment that is partially or fully executed manually		
3. Examples	In the cash application process: - User should consecutively make logins to several programs before start working on the cases		
4. Minimum data needed	Activities, start timestamps, end timestamps, resources		
5. Guideline on how to identify this IO	Expected output: <i>Steps 1-2</i> allow determining the list of fragments with the highest average processing time. <i>Step 3</i> allows compiling the list of all manual fragments with activities that have the highest processing time		
	#	Step	Explanation
	1	<p>1) Open event log in the process discoverer</p> <p>2) In the View section of Visualization settings, choose <i>Process map</i> and <i>Activities</i> perspective. Select <i>Duration</i> overlay and choose <i>Average</i></p>	<p>Result of the step: generated process map based on average duration and activities perspective</p> 
	2	From the process map, find sequences of activities that have high average processing time. List these fragments of activities	<p>Result of the step: list of fragments with the highest average processing time</p> 

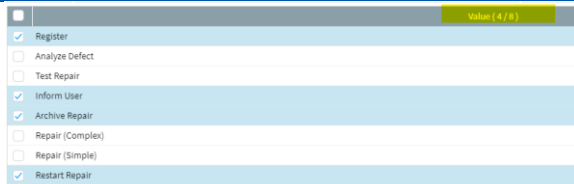
	3	<div><div><div>1) Open the <i>Filter log</i> screen.</div><div>2) In Case → <i>Attribute</i> filter, set Primary attribute as <i>Event attribute</i>, <i>Activity</i>; set Secondary attribute as <i>Event attribute</i>, <i>Resource</i>.</div><div>3) Select an activity from the fragment identified in <i>Step</i> 2.</div><div>4) Check the resources that perform this activity (is this activity performed by human or non- human resources (e.g., System)).</div><div>5) Repeat for all activities of the fragment.</div><div>6) List this fragment if all or most of its activities are</div></div></div>	<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div></div></div>	

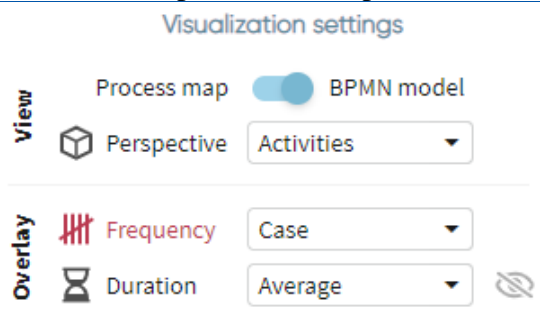
		performed by human resources.		
		Repeat for all fragments identified in <i>Step 2</i>		
	Output: List of severe manual fragments in terms of the highest processing time			
6. Redesign possibilities	- Process fragment automatization			
7. References				
7.1. References to the academic literature	[37, 38]			

Process-related improvement opportunities

1. Improvement opportunity (IO)	Manual process			
2. Definition	Process most activities of which are executed manually			
3. Examples	In the contract handling process: - Process is executed mostly manually, activities include opening a contract, assignment, research			
4. Minimum data needed	Activities, end timestamps, resources			
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allow determining if it is a manual process.			
	#	Step	Apromore example	Explanation
	1	<div>1) Open event log in the process discoverer</div> <div>2) Open the <i>Filter log</i> screen.</div> <div>3) In Case → <i>Attribute</i> filter, set Primary attribute as <i>Event attribute</i>, <i>Resource</i>.</div>		Result of the step: generated list of resources in the filter editor
2	Analyze list of resources obtained in <i>Step 1</i> : <div>1) If all of the resources are human (e.g., names and surnames but <i>NOT</i> System), conclude</div>		If process is performed only by human resources, it is a manual process. If there is one or several non-human resources, it is needed to determine for what percentage of activities they are accountable to decide whether it is a manual process (on the last	

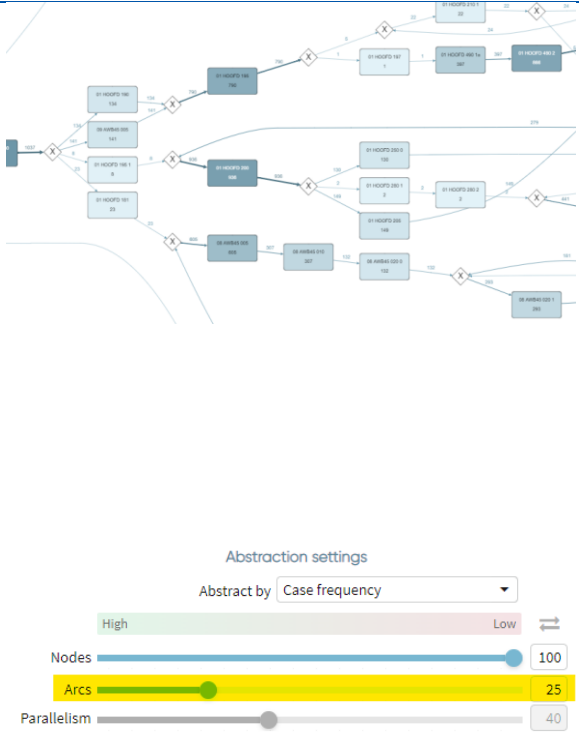
		<p>that this is a manual process</p> <p>2) If there is one or several non-human resources (e.g., System), set Secondary attribute as <i>Event attribute, Activity</i>. Select the first non-human resource. List activities performed by this resource. If there are more non-human resources, repeat for them listing activities that they are performing.</p> <p>Set Primary attribute as <i>Event attribute, Activity</i>. Disable Secondary attribute. Select all the listed above activities performed by non-human resources.</p>	 	<p>example screenshot, 50% activities are performed by non-human resources therefore it can be concluded that it is NOT a manual process).</p> <p>Result of the step: decision if this process is manual</p>
--	--	---	---	---

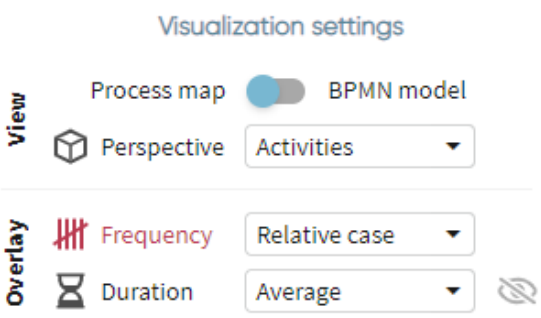
		<p>Check the number of activities performed by non-human resources out of all activities. If the percentage of activities performed by non-human resources is small, this is a manual process.</p>		
	Output: Decision if the process can be claimed as manual			
6. Redesign possibilities	<ul style="list-style-type: none">- Process automatization- Introduce technological solution that will diminish the need in manual task execution			
7. References				
7.1. References to the academic literature	[38]			

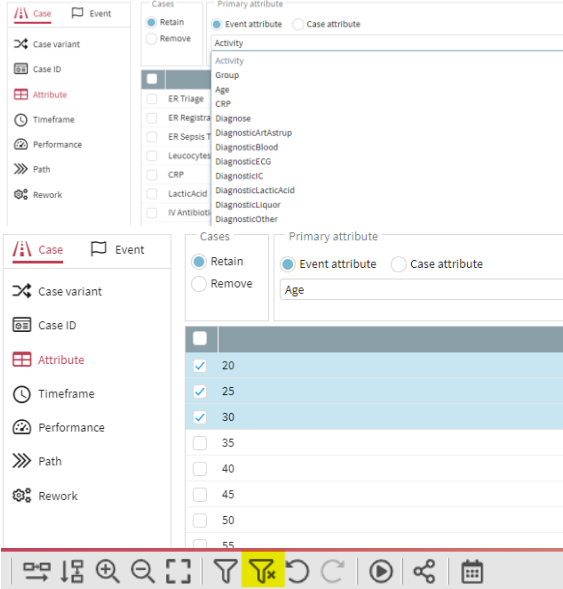
1. Improvement opportunity (IO)	High process complexity		
2. Definition	Process with high number of loops and/or branches (decision points)		
3. Examples	In the university matriculation renewal process: - Sophisticated control flow that increases cycle times and number of errors		
4. Minimum data needed	Activities, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allow determining how much complex the process is.		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the View section of Visualization settings, choose <i>BPMN model</i> and <i>Activities</i> perspective. Select <i>Frequency</i> overlay and choose <i>Case</i>	

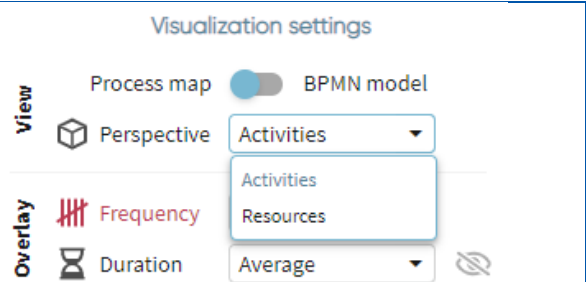
To better visualize decision points, BPMN model is used

Result of the step: generated BPMN model based on case frequency

	<div data-bbox="521 244 936 973"> <p>2</p> <ol style="list-style-type: none"> 1) Analyze process map to determine if it has extreme number of loops, and/or branches, and/or decision points (diamond shape symbol). 2) If there are many loops, and/or branches, and/or decision points, the process is complex. <p>If needed, increase the percentage of arcs in the Abstraction settings (see the screenshot) to have a more detailed overview.</p> </div> <div data-bbox="947 236 1523 973">  </div> <div data-bbox="1552 244 2040 311"> <p>Result of the step: decision on the complexity of the process</p> </div>
<p>6. Redesign possibilities</p>	<p>Output: Identified high or low process complexity</p> <ul style="list-style-type: none"> - Decompose the process by dividing it by several straightforward sub-processes
<p>7. References</p> <p>7.1. References to the academic literature</p>	<p>[39, 40]</p>

1. Improvement opportunity (IO)	Similar process variants		
2. Definition	Two or more process variants that are similar in terms of case execution for different types of case attributes or by different resources		
3. Examples	<p>In the management consulting process:</p> <ul style="list-style-type: none"> - Analysts with the same set of responsibilities analyses client's case by searching information on the Internet as the first step of the process. They are using different search engines and different keywords that leads to different analysis conclusions and providing different set of services later on to the client. 		
4. Minimum data needed	Activities, end timestamps, case attributes		
5. Guideline on how to identify this IO	Expected output: Steps 1-3 allow determining if process contains two or more process variants		
	#	Step	<div> <div> <div>Apromore example</div> <div>  </div> </div> <div>Explanation</div> </div>
	1	1) Open event log in the process discoverer 2) In the <i>Visualization settings</i> , select <i>Frequency overlay</i> and choose <i>Relative case</i> . In the <i>View</i> section, choose the <i>Activities</i> perspective	Result of the step: generated process map based on relative case frequency and activities perspective

	2	<ol style="list-style-type: none"> 1) Open the <i>Filter log</i> screen. 2) Using <i>Case</i> → <i>Attribute</i> filter, select specific event attribute based on which you would like to examine variants. 3) Select criteria for the first filtered process map (e.g. to compare how process differs for patients 20-30 y.o. and 60-70, firstly select 20-30 values). 4) Click OK. 5) Save generated process map. 6) Clear all filter criteria. 7) Open the <i>Filter log</i> screen. 8) Using <i>Case</i> → <i>Attribute</i> filter, select the same event attribute based on which 		<p>Result of the step: several process maps generated based on criteria</p>
--	---	--	--	--

		<p>you would like to examine variants.</p> <p>9) Select second criteria. Click OK.</p> <p>10) Save generated process map.</p> <p>If needed, generate more process maps based on criteria.</p> <p>Depending on the process, you might need to switch to different perspective or choose <i>Duration overlay</i>.</p>		
	3	<p>Compare generated in Step 2 process maps to define if some activities or sequences of activities have abnormal performance based on selected criteria. List these activities</p>		<p>Result of the step: list of activities and/or process fragments with abnormal variation that leads to formation of similar process variants</p>
	<p>Output: List of activities and/or process fragments with abnormal variation</p>			
6. Redesign possibilities	<ul style="list-style-type: none"> - Standardize similar processes to eliminate possibility of formation of similar process variants - Generalize the process by centralizing similar process variants 			
7. References				

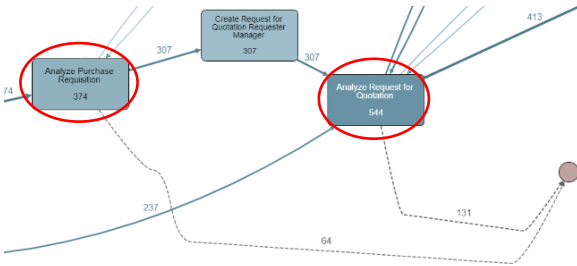
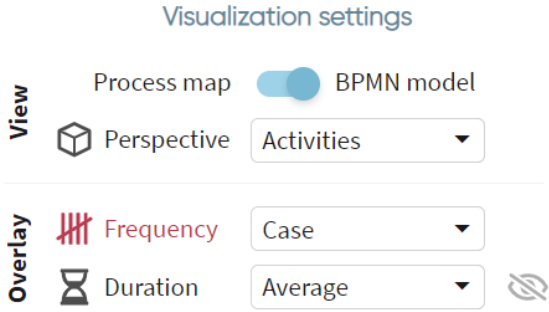



7.1. References to the academic literature

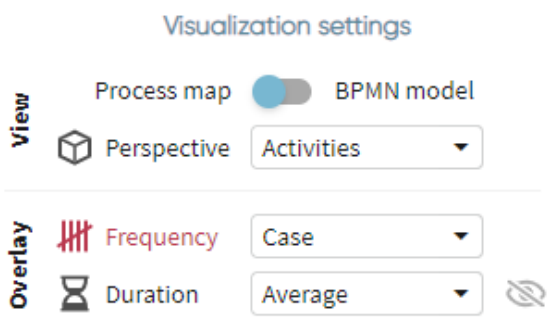
[41]

Process wastes-related improvement opportunities

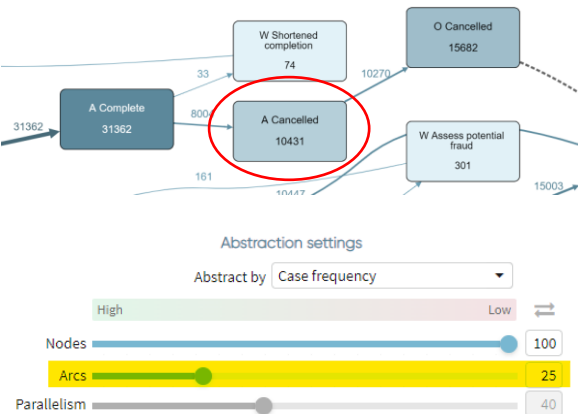
1. Improvement opportunity (IO)	Overprocessing		
2. Definition	Part of the business process, the output of which provides value neither to the client nor the business		
3. Examples	<p>In the loan application process:</p> <ul style="list-style-type: none"> - Loan offer documents are prepared, but the client cancels the offer - Several loan offer documents are prepared, but in the end, only one of them proceeds further <p>In the university matriculation process:</p> <ul style="list-style-type: none"> - Matriculation documents that are prepared are not needed because a student did not pay for their studies 		
4. Minimum data needed	Activities, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-4 allows identifying overprocessing. Step 5 gives an understanding of how severe identified cases are in terms of their frequency.		
	#	Step	<div> <div>Apromore example</div> <div> <div>Visualization settings</div> <div> <div>Process map</div> <div>BPMN model</div> </div> <div> <div>View</div> <div> <div>Perspective</div> <div>Activities</div> </div> </div> <div> <div>Overlay</div> <div> <div>Frequency</div> <div>Case</div> </div> <div> <div>Duration</div> <div>Average</div> </div> </div> </div> </div> <div>Explanation</div> <div> Result of the step: generated process map </div>

	<p>2</p> <p>From the process map,</p> <ol style="list-style-type: none"> 1) Find decision activities (check, verification, etc.) 2) Select only those decision activities that have knock-out effect (that have dashed arcs to the endpoint) and that occur when the final output of the process is not ready (at the stage when output do not provide value neither to the client or the business). List these activities 		<p>Focus on the decision activities that may result in skipping of other activities, this is one of the ways overprocessing manifests itself (on the example screenshot, these activities are circled in red; the process is ended on the analysis stage, when there is no output produced that will bring a value)</p> <p>Result of the step: list of activities that potentially indicate overprocessing</p>
	<p>3</p> <p>In the <i>Visualization settings</i>, switch to BPMN model</p>		<p>Result of the step: generated BPMN model</p>

	4	Examine the BPMN model to list parallel checks		<p>Parallel checks are one of the ways overprocessing manifests itself</p> <p>Result of the step: list of additional activities that potentially indicate overprocessing</p>
	5	Compare frequencies of identified overprocessing activities in <i>Steps 2 and 4</i> to select activities with the highest case frequencies		<p>The more frequent is overprocessing activity, the more severe is overprocessing</p> <p>Result of the step: final list of activities that contain overprocessing</p>
	Output: list of activities that contain overprocessing			
6. Redesign possibilities	<ul style="list-style-type: none"> - Relocate controls to earlier stages of the process to minimize or eliminate the production of the not needed output 			
7. References				
7.1. References to BPIC submissions	[28, 29, 42]			
7.2. References to the academic literature	[43]			

1. Improvement opportunity (IO)	Overproduction		
2. Definition	Executed business process instance, the output of which is not later used (output is higher than demand)		
3. Examples	In the automobile manufacturing process: <ul style="list-style-type: none"> - Due to inaccurate sales forecast, a large number of produced cars become an inventory, which led to high logistics costs 		
4. Minimum data needed	Activities, end timestamps		
5. Guideline on how to identify this IO	Expected output: Steps 1-2 allows identifying overproduction.		
	#	Step	Apromore example
	1	1) Open event log in the process discoverer 2) In the <i>Visualization settings</i> , select <i>Frequency overlay</i> and choose <i>Case</i> . In the <i>View</i> section, choose the <i>Activities</i> perspective	

Result of the step: generated process map based on case frequency and activities perspective

	<p>2</p> <p>From the process map,</p> <ol style="list-style-type: none"> 1) Find activities that indicate rejections, cancellations, etc., and before them the final output of the product is ready 2) List these activities and number of cases <p>If needed, increase the percentage of arcs in the Abstraction settings (see the second screenshot) to have a more detailed overview</p>		<p>One of the manifestations of overproduction is cancellations (on the example screenshot, circled in red; at this stage of the process, loan offer is finalized, all supporting documents are prepared but it is cancelled by the customer)</p> <p>Result of the step: list of activities that lead to overproduction and the number of cases affected</p>
<p>6. Redesign possibilities</p>	<p>Output: list of activities that lead to overproduction and the number of cases affected</p> <p>- NA</p>		
<p>7. References 7.1. References to BPIC submissions</p>	<p>[28, 29, 42]</p>		

List of references

- [1] Lashkevich, K., & Milani, F. P. (2020). Business process improvement opportunities: a framework to support business process redesign.
- [2] Sharma, A. S. (2021). Waste Identification from Event Logs.
- [3] Reijers, H. A., & Mansar, S. L. (2005). Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega*, 33(4), 283-306.
- [4] *Apromore Quick start*. (n.d.). Retrieved from <https://documentation.apromore.org/quickstart/index.html#>
- [5] Souza, A., Azevedo, L. G., & Santoro, F. M. (2017). Automating the identification of opportunities for business process improvement patterns application. *International Journal of Business Process Integration and Management*, 8(4), 252-272.
- [6] Thabet, D., Ayachi Ghannouchi, S., & Hajjami Ben Ghezala, H. (2018, September). A process mining-based solution for business process model extension with cost perspective context-based cost data analysis and case study. In *IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 434-446). Springer, Cham.
- [7] Martin, N., Depaire, B., & Caris, A. (2016). The use of process mining in business process simulation model construction. *Business & Information Systems Engineering*, 58(1), 73-87.
- [8] Mazz, M., & Kumar, M. (2012, December). Structured method for business process improvement. In *2012 Third International Conference on Services in Emerging Markets* (pp. 183-188). IEEE.
- [9] Milan, R., Milan, B., Marko, C., Jovanovic, V., Dalibor, B., Bojic, Z., & Avramovic, N. (2014). Implementation of business process reengineering in human resource management. *Engineering Economics*, 25(2), 211-222.
- [10] Falk, T., Griesberger, P., & Leist, S. (2013, June). Patterns as an artifact for business process improvement-insights from a case study. In *International Conference on Design Science Research in Information Systems* (pp. 88-104). Springer, Berlin, Heidelberg.
- [11] Niedermann, F., & Schwarz, H. (2011). Deep business optimization: Making business process optimization theory work in practice. In *Enterprise, business-process and information systems modeling* (pp. 88-102). Springer, Berlin, Heidelberg.
- [12] Hansen, J. (2013). *Analyzing Volvo IT Belgium's Incident and Problem Management Data Using Automated Business Process Discovery*. Retrieved from: <http://ceur-ws.org/Vol-1052/paper6.pdf>
- [13] Kang, C. J., Kang, Y. S., Lee, Y. S., Noh, S., Kim, H. C., Lim, W. C., Kim, J., & Hong, R. (2013). *Process Mining-based Understanding and Analysis of Volvo IT's Incident and Problem Management Processes*. Retrieved from: <http://ceur-ws.org/Vol-1052/paper8.pdf>
- [14] Teinemaa, I., Leontjeva, A., & Masing, K. O. (2015). *BPIC 2015: Diagnostics of Building Permit Application Process in Dutch Municipalities*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2015:bpic2015_paper_3.pdf

- [15] Panpanich, P., Porouhan, P., & Premchaiswadi, W. (2015). Analysis of handover of work in call center using social network process mining technique. In *2015 13th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering 2015)* (pp. 97-104). IEEE.
- [16] Yang, H., Park, M., Cho, M., Song, M., & Kim, S. (2014). A system architecture for manufacturing process analysis based on big data and process mining techniques. In *2014 IEEE international conference on big data (big data)* (pp. 1024-1029). IEEE.
- [17] Meyer zu Wickern, V. F. A., Juluru, M., Roy, A., Nguyen, T., & Vu, V. (2019). *Analysis and prediction of purchasing compliance using process mining*. Retrieved from: <https://icpmconference.org/2019/wp-content/uploads/sites/6/2019/07/BPI-Challenge-Student-Submission-4.pdf>
- [18] Augusto, A., Leno, V., & Reissner, D. (2019). *BPI Challenge 2019 Report: a Purchase-to-Pay Process Analysis*. Retrieved from: <https://icpmconference.org/2019/wp-content/uploads/sites/6/2019/07/BPI-Challenge-Student-Submission-1.pdf>
- [19] Kim, J., Ko, J., & Lee, S. (2019). *Business Process Intelligence Challenge 2019: Process discovery and deviation analysis of purchase order handling process*. Retrieved from: <https://icpmconference.org/2019/wp-content/uploads/sites/6/2019/07/BPI-Challenge-Student-Submission-3.pdf>
- [20] R'bigui, H., & Cho, C. (2017). Customer order fulfillment process analysis with process mining: an industrial application in a heavy manufacturing company. In *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence*, 247-252.
- [21] van der Aalst, W. M. (2001). Re-engineering knock-out processes. *Decision Support Systems*, 30(4), 451-468.
- [22] Arias, M., & Rojas E. (2013). *Volvo Incident and Problem Management Behavior Analysis*. Retrieved from: <http://ceur-ws.org/Vol-1052/paper1.pdf>
- [23] Hevia, J., & Saint-Pierre, C. (2013). *Analyzing Volvo information with Process Mining*. Retrieved from: <http://ceur-ws.org/Vol-1052/paper7.pdf>
- [24] vanden Broucke, S. K. L. M., Vanthienen, J., & Baesens, B. (2013). *Volvo IT Belgium VINST*. Retrieved from: <http://ceur-ws.org/Vol-1052/paper3.pdf>
- [25] Outmazgin, N., & Soffer, P. (2016). A process mining-based analysis of business process work-arounds. *Software & Systems Modeling*, 15(2), 309-323.
- [26] Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., & Matzner, M. (2022). Detecting temporal workarounds in business processes—A deep-learning-based method for analysing event log data. *Journal of Business Analytics*, 1-25.
- [27] Povalyaeva, E., Khamitov, I., & Fomenko, A. (2017). *BPIC 2017: Density Analysis of the Interaction With Client*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2017:bpi2017_winner_student.pdf
- [28] Berger, F. (2017). *Mining Event Log Data to Improve a Loan Application Process*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2017:bpi2017_paper_3.pdf

- [29] Jeong, D., Lim, J., & Bae, Y. (2017). *BPIC 2017 - Analysis of Loan Process*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2017:bpi2017_paper_20.pdf
- [30] Ganesha, K., Dhanush, S., & SM, S. R. (2017, March). An approach to fuzzy process mining to reduce patient waiting time in a hospital. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1-6, IEEE.
- [31] Lee, S. K., Kim, B., Huh, M., Cho, S., Park, S., & Lee, D. (2013). Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry. *Expert systems with applications*, 40(1), 83-95.
- [32] Adriansyah, A., & Buijs, J.C.A.M. (2012). *Mining Process Performance from Event Logs*. Retrieved from: <https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2012:adriansyah.pdf>
- [33] Brils, J.H.H., van den Elsen, N.A.F., de Priester, J., & Slooff, T.A. (2018). *Business Process Intelligence Challenge 2018: Analysis and Prediction of Undesired Outcomes*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2018:bpi2018_paper_15.pdf
- [34] Pakileva, A., Skvortsova, E., Zakoryuchkin, N., Tsaplin, S., & Zarubin, V. (2020). *Business Process Intelligence Challenge 2020: Investigating Business Trips Arrangement Process at the Eindhoven University of Technology (TU/e)*. Retrieved from: https://icpmconference.org/2020/wp-content/uploads/sites/4/2020/10/ICPM_2020_paper_124.pdf
- [35] Premchaiswadi, W., & Porouhan, P. (2015). Process modeling and bottleneck mining in online peer-review systems. *SpringerPlus*, 4(1), 1-18.
- [36] Caballero-Hernández, J. A., Dodero, J. M., Ruiz-Rube, I., Palomo-Duarte, M., Argudo, J. F., & Domínguez-Jiménez, J. J. (2018, September). Discovering bottlenecks in a computer science degree through process mining techniques. In *2018 International Symposium on Computers in Education (SIIE)* (pp. 1-6). IEEE.
- [37] Yao, J., Wencheng, W., Prabhakara, J., Watts-Englert, J., Simmons, I., Mongeon, M., & Tharayil, M. (2017, June). Crowdsourcing Workflow Optimization to Internal Worker Crowds. In *2017 IEEE International Conference on Cognitive Computing (ICCC)* (pp. 56-63). IEEE.
- [38] Shraideh, A. H., Camus, H. G., & Yim, P. G. (2009). Business Process Optimization by Workflow Analysis. In *CAINE* (pp. 108-113).
- [39] Chang, S. E., Chen, Y. C., & Lu, M. F. (2019). Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process. *Technological Forecasting and Social Change*, 144, 1-11.
- [40] Aboulaid, H., Jardini, B., Sedqui, A., Elkayl, M., Britel, M. R., Amri, M., & Lyahyaoui, A. (2016, May). Process re-engineering and success of integration projects of information technologies case study: process modeling of a cross docking platform of a car manufacturer. In *2016 3rd international conference on logistics operations management (GOL)* (pp. 1-6). IEEE.
- [41] Li, C., Reichert, M., & Wombacher, A. (2011). Mining business process variants: Challenges, scenarios, algorithms. *Data & Knowledge Engineering*, 70(5), 409-434.

- [42] Carvallo, A., Henning, C., Razmilic, D., Lopez, R. R., Lee, J., Fernandez, J. P., & Arias, M. (2017). *Applying Process Mining for Loan Approvals in a Banking Institution*. Retrieved from: https://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2017:bpi2017_paper_35.pdf
- [43] Verenich, I., Dumas, M., Rosa, M. L., Maggi, F. M., & Francescomarino, C. D. (2016, June). Minimizing overprocessing waste in business processes via predictive activity ordering. In *International Conference on Advanced Information Systems Engineering* (pp. 186-202). Springer, Cham.
- [44] Dumas, M. (2021). *Business Process Mining. Lecture 5: Conformance Checking*. Retrieved from https://courses.cs.ut.ee/LTAT.05.025/2021_spring/uploads/Main/Lecture5.pdf
- [45] Dumas, M. (2021b). *Business Process Mining Lecture 6: Process Performance Mining*. Retrieved from https://courses.cs.ut.ee/LTAT.05.025/2021_spring/uploads/Main/Lecture6.pdf