Generic Process Performance Analysis

Master Thesis

L.S. Hernandez Altamirano

Supervisors:

prof.dr.ir. W.M.P. van der Aalst
dr. M. Westergaard
dr. M. Comuzzi

Final version

Eindhoven, November 2011
Abstract

Business Process Management Systems (BPMS) have high relevance for supporting today's business process execution. However, BPM is supported by other areas such as Business Intelligence, which is useful for the analysis and presentation of business information. This business information is generally related to process performance metrics.

Additionally, when it is required to redesign operational processes in BPM, two trends can be identified: Straight Through Processing (STP) and Case Handling (CH). STP is related to the complete automation of a business process, i.e., handling cases without human involvement. CH addresses the problem of processes that are too variable or too flexible to be captured in a process diagram. The CH orientation is supported by case management systems. Those systems register the information about the process execution in system logs, which are a rich source for process analysis techniques like Process Mining.

Process Mining allows us to analyze the process performance by distilling a process model from an event log and comparing it with a predefined model of the process owner, e.g., “process performance analysis with Petri Nets”. However, when processes are variable, the identification of the process model from an event log can be difficult. To address this problem, Process Mining has a simpler alternative that enhances the process and retrieves information about basic performance metrics, e.g. frequencies, averages case duration, etc.

Additionally, in this master thesis, we discuss the definition and implementation of a solution that supports the enhancement of the process. This solution must allow us to perform a generic process performance analysis with a BI orientation that facilitates the visualization of the performance metrics obtained with the analysis. Finally, the implementation of this solution is based on the development of a prototype application that we call GPPA (Generic Process Performance Analysis).

Keywords: event log, process mining, business process management, case handling, business intelligence, process performance.
Preface

This master thesis is the result of my graduation project for the Master of Business Information Systems at Eindhoven University of Technology. The project was carried out within the group of Architecture of Information Systems (AIS) from the Mathematics and Computer Science department of the TU/e. Additionally, this project was developed in collaboration with the department of Research and Development from Océ Netherlands Venlo.

First of all I would like to thank Wil van der Aalst for providing me this opportunity and for his guidance during my master's project. I would also like to thank Michael Westergaard because without his continuous support, expertise, and guidance as well, I wouldn't have been able to get this far. Furthermore, my thanks to Jakob Klok, for his support and counseling during my time in Océ. Additionally, I would like to thank Joyce Nakatumba and Joos Buijs for their valuable feedback.

I would also like to thank all my colleagues at the department and especially, my friends Hector, Manuel, Ezequiel, Cesar, Gokul and Andrés; who were my second family here in The Netherlands.

Last but not least, a special thanks to my mother, two brothers, my girlfriend Nelleke and her family, because without their support and motivation during these two years, I would not have got this far.

Santiago Hernández

November, 2011
## List of Contents

List of Contents.............................................................................................................. iii
List of Tables.................................................................................................................. vi
List of Figures................................................................................................................ vii

1 Introduction............................................................................................................... 1
  1.1 Assignment Context............................................................................................... 2
  1.2 Business Process Management............................................................................ 2
  1.3 Problem Definition............................................................................................... 4
  1.4 Overview............................................................................................................... 5

2 Preliminaries.............................................................................................................. 7
  2.1 Business Process Management.......................................................................... 7
  2.2 Business Intelligence........................................................................................... 8
    2.2.1 BPM and BI..................................................................................................... 9
  2.3 Process Mining.................................................................................................... 10
    2.3.1 Overview....................................................................................................... 10
    2.3.2 Event logs..................................................................................................... 10
      2.3.2.1 MXML..................................................................................................... 10
      2.3.2.2 XES......................................................................................................... 11
      2.3.2.3 XES vs. MXML..................................................................................... 14
    2.3.3 Discovery....................................................................................................... 14
    2.3.4 Conformance................................................................................................. 14
    2.3.5 Enhancement................................................................................................. 14
  2.4 Discussion............................................................................................................. 15

3 Preliminary Analysis................................................................................................. 16
  3.1 Case Study 1: Sales department of Dutch company........................................ 17
    3.1.1 Process Description....................................................................................... 17
    3.1.2 Information Structure.................................................................................... 17
    3.1.3 Process Analysis............................................................................................ 19
    3.1.4 Research Questions....................................................................................... 20
  3.2 Case Study 2: Dutch Municipality...................................................................... 20
    3.2.1 Process Description....................................................................................... 21
    3.2.2 Information Structure.................................................................................... 21
    3.2.3 Process Analysis............................................................................................ 23
    3.2.4 Research Questions....................................................................................... 25
  3.3 BPA Critical Evaluation...................................................................................... 25
    3.3.1 BPA description............................................................................................. 25
    3.3.2 Case study 1 evaluation................................................................................ 26
    3.3.3 Case Study 2 evaluation............................................................................... 29
  3.4 Usability............................................................................................................... 30
  3.5 Findings............................................................................................................... 31

4 Meta-model definition............................................................................................. 33
  4.1 Meta Model Definition....................................................................................... 34
    4.1.1 Log............................................................................................................... 35
    4.1.2 Trace.............................................................................................................. 35
    4.1.3 Event.............................................................................................................. 35
## Generic Process Performance Analysis

### 4.5 Summary

#### 4.4 Report Generation

- **4.4.3 Results**
- **4.4.2 Index Section**
- **4.4.1 Cover and Summary**

### 4.3 Charts Visualization

- **4.3.2 Pie Chart**
- **4.3.3 Line Chart**
- **4.3.1 Bar Chart**

- **4.3.1.1 Simple Bar Chart**
- **4.3.1.2 Bar Chart with Series**
- **4.3.1.3 Bar Chart with extra continuous axes**

- **4.3.2.4 Date Comparator**
- **4.3.2.3 Date Filter**
- **4.3.2.1 General Filter**

- **4.3.3.4 Activity Working Time per Group**
- **4.3.3.3 Line Chart with extra continuous axes**
- **4.3.3.2 Line Chart with Series**
- **4.3.3.1 Simple Line Chart**

- **4.3.1.4 Group by Time Unit Static**
- **4.3.1.2 Group Numeric**
- **4.3.1.1 Group General**

### 4.2 Mappings Library

- **4.2.3.4 Activity Working Time per Group**
- **4.2.3.3 Activity Working Time**
- **4.2.3.2 Time Between**
- **4.2.3.1 General Measurement**

- **4.2.2.4 Date Comparator**
- **4.2.2.3 Date Filter**
- **4.2.2.1 General Filter**

- **4.2.3 Measurement mappings**
- **4.2.2 Filtering Mappings Library**
- **4.2.1 Grouping Mappings Library**

- **4.2.1.4 Group by Time Unit Static**
- **4.2.1.2 Group Numeric**
- **4.2.1.1 Group General**

### 4.1 Charts

- **4.1.8 Chart**
- **4.1.7 Axis**
- **4.1.6 Mapping**
- **4.1.5 Group**
- **4.1.4 Property**

- **4.1.8 Plot**

### 5 Solution Implementation

- **5.2.1.2 Group Numeric**
- **5.2.1.1 Group General**

### 4.2 Mappings Library

- **4.2.2.4 Date Comparator**
- **4.2.2.3 Date Filter**
- **4.2.2.1 General Filter**

- **4.2.3 Measurement mappings**
- **4.2.2 Filtering Mappings Library**
- **4.2.1 Grouping Mappings Library**

- **4.2.1.4 Group by Time Unit Static**
- **4.2.1.2 Group Numeric**
- **4.2.1.1 Group General**
List of Tables

Table 3.1: Process Order log definition................................................................. 18
Table 3.2: Process Event Permits log definition..................................................... 22
Table 3.3: BPA usability evaluation...................................................................... 31
Table 5.1: Java chart library selection criteria...................................................... 68
Table 5.2: Report Library Selection Criteria......................................................... 71
Table 6.1: GPPA usability evaluation.................................................................... 95
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The BPM life-cycle to compare Workflow Management and Business Process</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Research Goals</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>CSCW Spectrum</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>BI tool visualization example</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>MXML format</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>XES structure</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>XES example</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Goals – Preliminary Analysis</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>Order's trace XES example</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>Order process model</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Event Permits trace XES example</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>Event Permits process model</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>Basic Performance Analysis</td>
<td>26</td>
</tr>
<tr>
<td>3.7</td>
<td>Order cases accumulated duration in hours per month</td>
<td>27</td>
</tr>
<tr>
<td>3.8</td>
<td>Average working time of “Behandelen”</td>
<td>28</td>
</tr>
<tr>
<td>3.9</td>
<td>Standard deviation of the working time of “Behandelen”</td>
<td>28</td>
</tr>
<tr>
<td>3.10</td>
<td>Maximum working time of Behandelen</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Research Goals – Meta-model Definition</td>
<td>33</td>
</tr>
<tr>
<td>4.2</td>
<td>Meta-model for graphical representation of properties of logs</td>
<td>34</td>
</tr>
<tr>
<td>4.3</td>
<td>XES event property example of the process Event Permits</td>
<td>35</td>
</tr>
<tr>
<td>4.4</td>
<td>Group construction using the concept:name property</td>
<td>36</td>
</tr>
<tr>
<td>4.5</td>
<td>Filter mapping example using the org:resource parameter value “John”</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>Number of events per activity</td>
<td>38</td>
</tr>
<tr>
<td>4.7</td>
<td>Frequency of resources per age intervals</td>
<td>40</td>
</tr>
<tr>
<td>4.8</td>
<td>Frequency of Activities per time period</td>
<td>41</td>
</tr>
<tr>
<td>4.9</td>
<td>Frequency of events per month of the year</td>
<td>43</td>
</tr>
<tr>
<td>4.10</td>
<td>Line Chart extension to Time Series Chart</td>
<td>49</td>
</tr>
<tr>
<td>4.11</td>
<td>Cover and Summary</td>
<td>50</td>
</tr>
<tr>
<td>4.12</td>
<td>Index</td>
<td>50</td>
</tr>
<tr>
<td>4.13</td>
<td>Results</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Research Goals – Tool Implementation</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Solution Proposal</td>
<td>54</td>
</tr>
<tr>
<td>5.3</td>
<td>GPPA chart configuration panel</td>
<td>55</td>
</tr>
<tr>
<td>5.4</td>
<td>GPPA mapping construction</td>
<td>55</td>
</tr>
<tr>
<td>5.5</td>
<td>Group General (By Group) configuration panel</td>
<td>56</td>
</tr>
<tr>
<td>5.6</td>
<td>Group Numeric (By Interval) configuration panel</td>
<td>57</td>
</tr>
<tr>
<td>5.7</td>
<td>Group By Period (By Period) configuration panel</td>
<td>58</td>
</tr>
<tr>
<td>5.8</td>
<td>Group by Time Unit (By Time Unit) configuration panel</td>
<td>59</td>
</tr>
<tr>
<td>5.9</td>
<td>Group by Time Unit Static (By Time Unit Static) configuration panel</td>
<td>59</td>
</tr>
<tr>
<td>5.10</td>
<td>Generic Filter (By Selection) configuration panel</td>
<td>60</td>
</tr>
<tr>
<td>5.11</td>
<td>Numeric Filter (By Value) configuration panel</td>
<td>61</td>
</tr>
<tr>
<td>5.12</td>
<td>Date Filter (By Date) configuration panel</td>
<td>62</td>
</tr>
<tr>
<td>5.13</td>
<td>Date Comparator(Between Dates) configuration panel</td>
<td>63</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.14</td>
<td>Measurement mappings configuration panel</td>
<td>63</td>
</tr>
<tr>
<td>5.15</td>
<td>General Measurement (By Measure) mapping configuration panel</td>
<td>64</td>
</tr>
<tr>
<td>5.16</td>
<td>Time Between mapping configuration panel</td>
<td>65</td>
</tr>
<tr>
<td>5.17</td>
<td>AWT mapping configuration panel</td>
<td>65</td>
</tr>
<tr>
<td>5.18</td>
<td>AWTG mapping configuration panel</td>
<td>66</td>
</tr>
<tr>
<td>5.19</td>
<td>Chart configuration</td>
<td>67</td>
</tr>
<tr>
<td>5.20</td>
<td>Activities frequency versus working time</td>
<td>69</td>
</tr>
<tr>
<td>5.21</td>
<td>Frequency of activities per resource</td>
<td>70</td>
</tr>
<tr>
<td>5.22</td>
<td>GPPA Cover</td>
<td>73</td>
</tr>
<tr>
<td>5.23</td>
<td>GPPA Summary</td>
<td>74</td>
</tr>
<tr>
<td>5.24</td>
<td>GPPA Index</td>
<td>75</td>
</tr>
<tr>
<td>5.25</td>
<td>GPPA Results</td>
<td>75</td>
</tr>
<tr>
<td>6.1</td>
<td>Research Goals – Case Studies Evaluation</td>
<td>77</td>
</tr>
<tr>
<td>6.2</td>
<td>Resources' participation for process Order</td>
<td>78</td>
</tr>
<tr>
<td>6.3</td>
<td>Activities' participation for process Order</td>
<td>79</td>
</tr>
<tr>
<td>6.4</td>
<td>Number of cases created and closed per month</td>
<td>80</td>
</tr>
<tr>
<td>6.5</td>
<td>Average, STD and Max Case duration</td>
<td>81</td>
</tr>
<tr>
<td>6.6</td>
<td>Average working time of Behandelen</td>
<td>82</td>
</tr>
<tr>
<td>6.7</td>
<td>Standard deviation of the working time of Behandelen</td>
<td>83</td>
</tr>
<tr>
<td>6.8</td>
<td>Maximum working time of Behandelen</td>
<td>84</td>
</tr>
<tr>
<td>6.9</td>
<td>Time Between “Behandelen” and “Order kan ingevoerd worden”</td>
<td>85</td>
</tr>
<tr>
<td>6.10</td>
<td>Number of Cases with delay per year</td>
<td>86</td>
</tr>
<tr>
<td>6.11</td>
<td>Number of Cases with delay per year per main handler</td>
<td>87</td>
</tr>
<tr>
<td>6.12</td>
<td>Resources's participation for process “Event Permits”</td>
<td>88</td>
</tr>
<tr>
<td>6.13</td>
<td>Document types participation for process “Event Permits”</td>
<td>89</td>
</tr>
<tr>
<td>6.14</td>
<td>Number of cases created per month</td>
<td>90</td>
</tr>
<tr>
<td>6.15</td>
<td>Time Between Aanvraag vergunning and the completion of Vergunning</td>
<td>91</td>
</tr>
<tr>
<td>6.16</td>
<td>Time Between Ontvangstbevestiging and the completion of Vergunning</td>
<td>92</td>
</tr>
<tr>
<td>6.17</td>
<td>Cases with a time over the standard between Ontvangstbevestiging and Vergunning.</td>
<td>93</td>
</tr>
<tr>
<td>A.1</td>
<td>XES extensions with attributes</td>
<td>101</td>
</tr>
<tr>
<td>A.2</td>
<td>Event type association</td>
<td>102</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Business process management systems have emerged during the last decades as a new technology to support, enact and control business processes. They usually record information about the execution of process activities in event logs. Those logs can be analyzed with methods such as Process Mining, which aims to “discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's (information) systems.” [1]. Furthermore, the discovery of a structured process model is not an easy task when processes with a high number of activities and a high number of exceptions or deviations (e.g. alternative paths on the process execution) are taken into consideration.

Additionally, if the structured process cannot be discovered with flexible processes, we can use a simpler approach, which enhances the process and monitors it's performance by detecting basic performance metrics [2], e.g. flow times, waiting times, processing times, resource utilization, etc. It is also possible to analyze performance characteristics based on the properties of a case or a process instance, e.g. expensive cases requires less processing. Furthermore, this performance information is useful for detecting wrong work distributions, undesired case and activities durations, and case rules violations (e.g. cases that delayed).

Finally, the objective of this master thesis project is to investigate an alternative for applying this simpler approach, that we have labeled as Generic Process Performance Analysis (GPPA). Furthermore, this will be developed within a Business Intelligence (BI) orientation [3] supported by the use of charts, which are able to show performance results in an easy to read and understandable way [4].

This chapter introduces the alternative mentioned above in more detail. In Section 1.1 the assignment context is explained. Section 1.2 gives a short description of some preliminary concepts. Section 1.3 discusses the problem definition and Section 1.4 gives an overview of the master thesis with an explanation of the solution approach and the outline of the thesis.
1.1 Assignment Context

This master thesis is the result of my graduation project for the master of Business Information Systems at the Eindhoven University of Technology (TU/e). This project is carried out within the group of Architecture of Information Systems (AIS) from the Mathematics and Computer Science department of the TU/e. The AIS group investigates methods, techniques and tools for the design and analysis of process-aware information systems (e.g. workflows). Additionally, the AIS group also tries to model and analyze the business processes and organizations they support, and it is probably one of the best Business Process Management (BPM) group around the world [5].

Additionally, this project was developed in the department of Research and Development from Océ N.V., which is a Netherlands-based company founded in 1877 and acquired by Canon of Japan in 2010. Océ is active in over 100 countries and employs 20,000 people worldwide. It focuses on the development, manufacturing and selling of printing-copying hardware and related software [6]. One of Océ software solutions is DossierFlow, which is a web based BPM solution with Case Handling (CH) [7] orientation that supports digital filing and processing of case files.

DossierFlow has been used in different environments like commercial companies and government institutions. Furthermore, Océ is interested in applying Process Mining techniques in some DossierFlow’s databases obtained from its customers. This is done with the purpose of extracting process performance information based on the activities and cases’ durations, number of cases done in the past, work distributions, process rules validations and possible bottlenecks detection. However, Océ requires that the process performance information must be presented in a graphical way with a BI orientation (e.g. use of charts).

1.2 Business Process Management

In this section some preliminary concepts are addressed, such as: Business Process management, Business Process Management Systems, Business Intelligence, Case Handling, Process Mining and ProM.

Many definitions have appeared in relation to the term Business Process Management (BPM) [8–11], we use the one defined by Van der Aalst, Ter Hofstede, and Weske [8] that defines BPM as an approach that “includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes.”. Additionally, BPM can be considered as an extension of classical Workflow Management (WFM) systems and approaches. This relation between BPM and WFM is presented in Figure 1.1 that shows the BPM life-cycle with its four phases: process design, system configuration, process enactment, and process diagnosis. And the support of WFM in the process design phase.
CHAPTER 1. INTRODUCTION

Additionally, Business Process Management Systems (BPMS) are defined as “generic software systems that are driven by explicit process designs to enact and manage operational business processes” [7]. Although, BPM is related with other areas such as Business Process Analysis (BPA) that covers aspects omitted by traditional workflow products (e.g., diagnosis, simulation, etc.) and is supported by emerging areas like Business Activity Monitoring (BAM) [12], [13]. This area is considered as an extension of the BI approach, which refers to “technologies, applications and practices for the collection, integration, analysis and presentation of business information”, which is useful for detecting process performance metrics in BPMS [14].

When it is required to redesign operational processes in BPM, two trends can be identified: Straight Through Processing (STP) and Case Handling (CH) [8]. STP is related to the complete automation of a business process, i.e., handling cases without human involvement. CH addresses the problem of processes that are too variable or too flexible to be captured in a process diagram [2]. Furthermore, in CH it is possible to model the normal route of a case by creating data-driven workflows rather than process-driven ones, and allow authorizations that skip or undo activities [8]. The focus is on the case as a whole rather than on individual work-items distributed over work-lists [7].

Once BPM has been defined, we address the concept of Process Mining [2]; this is in relation to the support that it brings to BPM for process analysis. Process Mining is “A relatively young research discipline that sits between computational intelligence and data mining on one hand, and process modeling and analysis on the other hand.” [1]. The goal of Process Mining is to “discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's (information) systems” [1]. Furthermore, the three main types of Process Mining are: discovery (e.g. process discovery), conformance (e.g. process validation) and enhancement (e.g. performance metrics detection and analysis) [1]. In this master thesis context, Océ is interested in determining performance metrics (enhancement) of the processes supported by DossierFlow, e.g. frequencies, times, resource utilization, and variability.

Finally, to apply Process Mining we need tool support; some of the most relevant options are [15]:

![Figure 1.1: The BPM life-cycle to compare Workflow Management and Business Process Management [8].](image)
ProM\(^1\) and Futura\(^2\). In this master thesis we have selected ProM, which is a tool developed by the AIS group of the Tu/e that allows us to apply the three main types of Process Mining: *discovery*, *conformance* and *enhancement* of processes. It also has a wide stock of implemented mining algorithms and is available under an Open Source license that gives us the opportunity to implement and add new plug-in solutions.

### 1.3 Problem Definition

As discussed in the previous sections, BPM is an approach that supports the design, enactment, management, and analysis of operational business processes \([8]\). Furthermore, we could see that BPM is supported by BPMS, which records the information related to the process execution in event logs. Additionally, process owners are interested in analyzing the performance of their processes, which requires a method that allows them to extract knowledge from the event logs. Here, is where Process Mining appears as a solution for it and we could realize that two main options exist for applying it: *process oriented* and *data oriented*. The first one is related with process conformance validations which analyze if the current process sequence definition is being followed in practice. The second one, retrieves general performance information like flow times, waiting times, processing times, resource utilization, etc.

Additionally, we also could observe that obtaining a structured process model can be complex in flexible environments, like in processes supported by CH. Those processes have more than one way for executing a case or a process instance. Furthermore, a simpler approach is selected for these types of flexible processes, which is based on the detection of *general performance metrics in a generic way (e.g. considering the event logs properties)* with a BI orientation. This will allow us to apply two specific tasks: detection of basic performance metrics and analysis of performance characteristics.

The approach will be validated with the logs of DossierFlow, which is a BPM solution with a CH orientation. Furthermore, for applying Process Mining the ProM framework was selected. Though, the BI orientation of the approach is related to the visualization of the process performance information in an easy way to read (e.g. performance charts), which can be included in a consolidated performance report \([14]\). Additionally, this report must be configurable (e.g. titles and text inserted by customers) and available in a standard portable document format (e.g. PDF report).

Finally, the current version of ProM (ProM 6) doesn't have a plug-in able to apply a *generic process performance analysis* based on the event log data and attributes. Additionally, there is not a plug-in available to export performance information in a standard portable document format. Based on the initial analysis, we define the following two problem statements:

*Problem Statement 1*: There is not a plug-in available in ProM 6 that allows the execution of a generic process performance analysis with a BI orientation.

*Problem Statement 2*: There is not a plug-in available in ProM 6 that allows the generation of a consolidated report with information obtained throughout a generic process performance analysis with chart support.

---

\(^1\) See http://www.promtools.org/prom6/
\(^2\) See http://www.futuratech.nl/site/index.php
1.4 Overview

To solve the problem statements, it's necessary to implement a new ProM plug-in solution that will be labeled as GPPA (Generic Process Performance Analysis), which should be able to perform a generic process performance analysis based on the event log data and attributes. Furthermore, this plug-in solution must also be able to generate a report that contains the information gathered from the performance analysis (e.g. charts with Meta information defined by the user). Additionally, to generate the chart performance graphics, it is necessary to define a Meta-model that addresses the properties available in an event log in a generic way. This definition will allow us to explain the methods that will be used to display basic performance metrics, e.g. average activity duration per year.

Additionally, the new plug-in is validated with two case studies; the first one is a Dutch municipality and the second one is the sales department from a Dutch company. Both of them are DossierFlow customers. Furthermore, a preliminary analysis will be done with these two case studies, in order to obtain the desired performance metrics that they want to see on the new plug-in, and that cannot be gathered with current ProM available solutions, e.g. the Basic Performance Analysis plug-in from ProM 5.

Finally, to summarize the steps to perform in this master thesis project, a solution approach is presented in Figure 1.2, which shows the four research goals defined for this project.

---

Figure 1.2: Research Goals.
Once the research goals have been defined, the next step is to define the method for addressing them, which is composed of the following steps:

- **Perform a literature study on relevant concepts:** the first step is to gather information from existing literature on topics like: BPM, BI, Process Mining and Event log formats (XES and MXML). The findings of these topics are presented in Chapter 2.

- **Conduct a preliminary analysis:** a preliminary analysis has to be made for gathering basic performance metrics required from both case studies, and also for evaluating an available solution (BPA ProM 5). This is presented in Chapter 3.

- **Define the Meta-model and implement the GPPA:** once performance metrics and the strong points and gaps of the existing solution (BPA ProM 5) have been defined. The next step is to define the Meta-model for performing a *generic process performance analysis*. This is presented in Chapter 4.

- **Implement the GPPA:** to implement a plug-in solution in ProM for performing a *generic process performance analysis*. This is presented in Chapter 5.

- **Evaluation of the solution approach through the case data:** after the implementation of the GPPA, the next step is to validate it with the two study cases. This will be presented in Chapter 6.

- **Conclude:** Finally, a conclusion about the work done will be made, analyzing the GPPA strong points, limitations, and future work to improve it. This will be presented in Chapter 7.
Chapter 2

Preliminaries

This chapter introduces preliminary concepts used throughout this master thesis. Section 2.1 starts with the description of BPM and CH. Section 2.2 describes BI in relation to BPM, Section 2.3 introduces the concept of Process Mining considering its different types, and the event log definition. Finally, in Section 2.4 a brief discussion is mentioned about the concepts determined above, in relation to the solution proposed to fix the problem stated in the previous chapter.

2.1 Business Process Management

The area of Business Process Management (BPM) has received considerable attention due to its potential for increasing productivity and saving costs [16]. Furthermore, BPM uses the concept of a process as a starting point for understanding how a business operates, and how it can be improved. Additionally, many definitions have appeared in relation to BPM, we will use one done by Van der Aalst, Ter Hofstede, and Weske [8] that defines BPM as an approach that “includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes”.

Additionally, BPM can be considered as an extension of the classical Workflow Management (WFM) systems and approaches, because it aims to support operational business process by applying Business Process Automation (BPA), which sometimes is referred to as WFM [16]. This relation was presented in Figure 1.1 from Chapter 1, which shows the BPM life-cycle in comparison with Workflow Management. Furthermore, business processes descriptions (e.g. process model) are useful for guiding the performance of business activities, because it allows us to distribute the work to resources and also to track the progress of a particular process instance [16]. The progress tracking can be achieved by recording the execution of process events in log files, which is a rich source of information for process analysis, e.g. Process Mining.

However, WFM requires structured processes descriptions, which are based on a Straight Through Processing (STP) [8]. This requirement of a structured process is related to one of the goals of WFM, which is to achieve business process automation [16]. Additionally, in real practice, this structured process description is difficult to achieve, when we consider exceptions or deviations in the process definition. This is common in environments that require flexibility in the process execution. Additionally, a new paradigm to support BPM is required, which is Case Handling CH [7], [8].
CHAPTER 2. PRELIMINARIES

CH addresses the problem of processes that are too variable or too flexible to be captured in a process diagram [2]. Furthermore, in CH it is possible to model a generic route of a case by using data-driven workflows, rather than process-driven ones, and allow for authorizations to skip or undo activities. The focus is on the case as a whole rather than on individual work-items distributed over work-lists [7]. Additionally, the process description made in CH is situated in the middle, between a structured and an unstructured process definition, and also between a data-centric and a process-centric approach. This is illustrated in Figure 2.1 that situates the CH paradigm in terms of the CSCW (Computer Supported Cooperative Work) spectrum.

Finally, BPM requires the support of tools like Business Process Management Systems (BPMS) which are defined as “generic software systems that are driven by explicit process designs to enact and manage operational business processes” [8]. However, BPM is related with other areas such as Business Process Analysis (BPA) that covers aspects omitted by traditional workflow products (e.g., diagnosis, simulation, etc.) and that are supported by emerging areas like Business Activity Monitoring (BAM) [12], [13]. Furthermore, BAM is considered as an extension of the BI approach.

2.2 Business Intelligence

The term Business Intelligence (BI) is related to “technologies, applications and practices for the collection, integration, analysis and presentation of business information” [14]. We can use BI for different uses, such as: “reporting, on-line analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining and predictive analytics” [18]. Furthermore, we focus in how BI can support BPM.
2.2.1 BPM and BI

BPM is related with the methods, techniques and tools to support the design, enactment, management and analysis of operational business processes [8]. Furthermore, BPM implies an understanding of the processes goals, the needs and skills of the people involved in the execution of the process, process changes, and possible improvement points [19]. To achieve this understanding it's necessary to analyze the process behavior, which can be related to the process performance.

The process performance is based on the detection of basic metrics, e.g. flow times, waiting times, processing times, resource utilization, etc. [2] Those metrics are useful for validating if the process goals are achieved, and also for knowing how the resources are dealing with them, e.g. resources versus delayed cases. Additionally, this information can be extracted from an even log by the use of BI analytics, which gives us different options and perspectives to analyze the data obtained, e.g. key performance indicators (KPIs), executive dashboards and consolidated performance reports [19]. Furthermore, BI allows us to make definitions of aggregate data with real (e.g. obtained from event log) process execution information, e.g. a performance chart that shows the number of process cases done by resource John in 2011.

To illustrate how BI shows the performance data, Figure 2.2 shows some visualizations used in BI tools (e.g. bar charts, pie charts, time series charts, etc.). The figure shows an example of the Jaspersoft tool.

![Figure 2.2: BI tool visualization example.](http://intranetblog.blogware.com/jaspersoft%20dashboard%20BI.jpg)
CHAPTER 2. PRELIMINARIES

2.3 Process Mining

Once the concepts of BPM and BI have been covered, we will now focus on Process Mining.

2.3.1 Overview

Process Mining is “a relatively young research discipline that sits between computational intelligence and data mining on one hand, and process modeling and analysis on the other hand” [1]. The goal of process mining is to “discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's (information) systems” [1]. Furthermore, the three main types of Process Mining are: discovery (e.g. process discovery), conformance (e.g. process validation) and enhancement (e.g. performance metrics detection and analysis).

Additionally, Process Mining is supported by different tools [15] such as: ProM and Futura. In this master thesis we have selected ProM, which is a tool developed by the AIS group of the Tu/e that allows the discovery, conformance and enhancement of processes. It also has a wide stock of implemented mining algorithms and is available under an Open Source license, which allows us to implement new plug-ins solutions.

Finally, an event log is required for applying Process Mining, which must have a basic structure that allows us to identify processes, cases or process instances, activities, resources and timestamps. Furthermore, this basic structure is associated with an event format, which in the case of ProM, there are two available options: MXML [20] and XES [21].

2.3.2 Event logs

The event log records all the events captured by a system (e.g. WFM, BPM, CH, etc.) during the case or process instance execution, which is a rich source of information for process analysis (e.g. Process Mining). It has information about the process activity done, the resource that performed it and the time when it happened. Additionally, these events can be ordered by their execution time and also grouped in relation to a process instance, which is defined as the trace of events [22]. Those traces of events give valuable information about how the process is actually behaving from a system perspective.

However, systems have different ways to capture the event information and this implies the need of defining a standard log format, which must be independent from the system structure and useful for process analysis purposes (e.g. events with activity, resource and time information). Examples of these standard formats are MXML and XES.

2.3.2.1 MXML

MXML is an event log format that was created in 2005 by Van Dongen and Van der Aalst [20], which has the purpose of standardizing the way that event logs are structured. An example of this format is shown in Figure 2.3, which demonstrates the general structure of the MXML format. This structure is composed by tags such as Process Instance to identify the process instance that is being
recorded, \textit{AuditTrailEntry} to describe the occurrence of a particular event, \textit{WorkflowModelElement} to identify the activity executed and \textit{EventType} to describe the status of this activity. This status is related with the activity lifecycle, e.g. “start” when the activity begins and “complete” when the activity is finished. Additionally, the tag \textit{Originator} is used for tracking the resource that performed the activity, and \textit{Timestamp} to determine the time when it happened. Finally, it is possible to have extra meta-data associated with an instance or with an event, e.g. the deadline of a particular activity. For that purpose we can use the \textit{Data} and \textit{Attribute} tags.

2.3.2.2 XES

In the past, the MXML format was used as the main format for applying Process Mining with the support of ProM, but this format has limitations in the way that it handles extra meta-data, because it is not possible to distinguish the type of data associated with it. For example, in the case of the \textit{deadline attribute} used in Figure 2.3, it is not possible to detect the type \textit{Date} associated with it. Furthermore, the distinction of this data type could be useful if we want to track the number of activities that had a delay.

To resolve this limitation the XES (eXtensible Event Stream) \cite{21} format was created, which allows us to distinguish the basic types such as \textit{String, Integer, Real, Date and Boolean}, in an event property. Additionally, XES keeps the classical notion of timestamps, resources and activities, which are necessary for a process performance analysis.

The structure of XES \cite{22} is shown in Figure 2.4, which considers the following elements:

- \textit{Log}: contains all the event information that is related to one specific process.

- \textit{Traces}: a log contains an arbitrary number of trace objects. Each trace describes the execution of one specific instance or case, of the logged process.

- \textit{Event}: traces contain an arbitrary number of event objects. Events represent atomic granules of activity that have been observed during the execution of a process. As such, an event has no duration.
CHAPTER 2. PRELIMINARIES

● **Attributes:** Logs, traces and events in XES don’t contain information by themselves. Furthermore, all the information in an event log is stored in attributes and there are five types of attributes, each one of them relates to a type of data value representation:

  - **String:** hold literal information which is generally untyped and of arbitrary length.
  - **Date:** hold information about a specific point in time (with milliseconds precision).
  - **Int:** hold a discrete integer number (with 64bit long precision).
  - **Float:** hold a continuous floating-point number (with 64bit double precision).
  - **Boolean:** hold a Boolean value which can be either true or false.

● **Extensions:** Attributes are a strong option for generating richer logs, but the XES standard does not define a specific set of attributes per log, trace, or event. This could be ambiguous when the interpretation of data is required. In order to address this, XES uses extensions that define a set of attributes on any level of the XES log hierarchy, e.g. log, trace, event and Meta for nested attributes. Additionally, XES has standard extensions that are vital for defining classic concepts in Process Mining: cases, activities, users, and time of an event. These main standard extensions are:

  - **Concept Extension:** defines an attribute which stores the generally understood name of type hierarchy elements, for all levels of the XES type hierarchy.
  - **Lifecycle Extension:** specifies for events, the life-cycle transition they represent in a transactional model of the activity involved.
  - **Organizational Extension:** it is useful for domains, where events can be caused by human actors, who are somehow part of an organizational structure. This extension covers notions such as: resource, role and group.
  - **Time Extension:** defines the exact date and time at which events were recorded. Recording timestamps for events is important, since this constitutes crucial information for many event log analysis techniques and especially for a process performance analysis.

These extensions are described in further detail in Appendix A.
After seeing the XES structure, the use of this format can be illustrated in Figure 2.5, which shows the use of the trace tags to identify the process instance or case in execution, event for the events detection, string to determine the activity and resource involved, and date for the timestamp and the deadline of activity. Additionally, the figure also shows the use of extensions associated with these Meta elements, which makes it possible to determine the activity, resource and timestamps, in a similar way done in MXML.

Figure 2.4: XES structure [22]
2.3.2.3 XES vs. MXML

After seeing how XES and MXML handle the information of the event logs, we could see that even though MXML allows the definition of extra meta-data related to an event or a case execution; it is not possible to determine the type of data. This reduces the opportunity of a deeper performance analysis. Because we can use the *deadline* attribute and determine the activities that had delay. Or even more, to use an *Integer* or *Real* value for gathering statistical information. For example, if an attribute related with costs is associated with the execution of an activity, it is possible to determine the average cost of its execution. Furthermore, due to these data type distinction capabilities, XES format has been selected as the standard one for the implementation of the GPPA.

2.3.3 Discovery

The *discovery* Process Mining type is a discovery technique that takes an event log and produces a model without using any a-priori information. Additionally, Process discovery is the best-known process mining technique [1].

2.3.4 Conformance

The *conformance* Process Mining type is a technique that can be used to check if the log really conforms to the model and vice versa [1]. Additionally, different types of models can be considered such as procedural models, organizational models, declarative process models, business rules, etc.

2.3.5 Enhancement

This type of Process Mining technique extends or improves an existing process model, using information about the actual process recorded in some event log [1]. For instance, by using timestamps in the event log we can extend the model to show bottlenecks, service levels, throughput times, and frequencies.

Additionally, in this master thesis project we will focus on this type of Process Mining technique, because it is useful for giving information about the process performance at a glance [23].

---

Figure 2.5: XES example.
2.4 Discussion

In this chapter, we could see the importance of the BPM area, which aims “to support the design, enactment, management, and analysis of operational business processes” [8]. Furthermore, BPM is applied throughout BPMS, which are able to record the process execution in event logs. Those logs are a rich source of information for process performance analysis, which requires a technique able to extract the knowledge involved in the events captured in a system log. Additionally, Process Mining is the technique that is able to perform this process analysis, and it's classified in three main types: discovery, conformance and enhancement.

We could also see, as well that processes descriptions are rather unstructured when we take into consideration processes that have flexibility for the activities executions, e.g. Case Handling. Additionally, the enhancement of the process is the most proper technique for this type of processes, which requires a simpler but not less effective approach that captures basic performance metrics from the process (e.g. flow times, waiting times, processing times, resource utilization, etc.). Furthermore, we can apply this approach with a BI orientation (e.g. using charts), which will increase the understanding of the metrics captured in the event logs. Additionally, the approach must be generic enough for allowing us to define attached views of the event properties, e.g. average case duration per month.

Finally, considering that ProM will be framework to support this generic approach, we have decided to implement a new plug-in solution not available in the current version of ProM (ProM 6), which is labeled as Generic Process Performance Analysis (GPPA).
Chapter 3

Preliminary Analysis

This chapter presents a preliminary analysis (please follow Figure 3.1) of the basic performance metrics of two processes from two DossierFlow system customers selected by Océ. Furthermore, we perform this analysis in the context of Process Mining. The description of these customers with their processes, information structure and performance questions are made in Section 3.1 and Section 3.2. Additionally, a critical evaluation of a previous solution, which is the Basic Performance Analysis (BPA) plug-in (only available for ProM 5) is made in Section 3.3. We perform this evaluation to identify the strong and weak points of the BPA solution and to validate the solution proposed in previous chapters of implementing the GPPA plug-in. Finally, in Section 3.4 the findings of the chapter are discussed.

Figure 3.1: Research Goals – Preliminary Analysis.
3.1 Case Study 1: Sales department of Dutch company

The first case study is about the sales department of a Dutch company, which is in charge of processes related with sales management of products, e.g. credit checks, complains handling, management of clients, etc. Furthermore, this DossierFlow customer has requested an analysis of the process Order.

Additionally, we received the DossierFlow database from this customer, containing information about the process execution during the time period of May 2010 to March 2011.

3.1.1 Process Description

The process Order is related to the registering and processing of purchase orders received from customers of this Dutch company. Additionally, this company has defined activities in relation to the status of a case in Order, and these are:

- **Behandelen**: order is received.
- **Order kan ingevoerd worden**: order is cleaned.
- **Order is gechecked, dossier archiveren**: order is processed.
- **Order kan gechecked worden**: order can be processed.
- **Order is bekend**: order is known.

3.1.2 Information Structure

To perform the Process Mining analysis with ProM, we need to extract the information from DossierFlow database tables and construct an even log using one of the two log formats that ProM supports: XES or MXML. Furthermore, we have selected XES, because of its capabilities of associating data types (e.g. String, Integer, Real, Boolean and Date) with the event attributes, and because ProM 6 uses it by default. For the BPA evaluation, ProM 6 has a feature for exporting the event log from XES to MXML.

Plus, DossierFlow handles the information related with the process and resources in two database tables. The fields selection criteria and the data type associated to them are shown in Table 3.1.
Additionally, in Figure 3.2 we can see an example of a trace from process Order using the XES format. (please continue to the next page)

<table>
<thead>
<tr>
<th>Field in DossierFlow Database</th>
<th>Type</th>
<th>Log Element</th>
<th>XES attribute associated</th>
<th>Selection Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>String</td>
<td>Event</td>
<td><code>org : resource</code></td>
<td>Gives information for the resource identification.</td>
</tr>
<tr>
<td>Main Handler Name</td>
<td>String</td>
<td>Event</td>
<td><code>mainHandler</code></td>
<td>Gives information for the role identification.</td>
</tr>
<tr>
<td>DSP Id</td>
<td>String</td>
<td>Trace</td>
<td><code>processId</code></td>
<td>Gives information for the process identification.</td>
</tr>
<tr>
<td>DSP Description</td>
<td>String</td>
<td>Trace</td>
<td><code>processName</code></td>
<td>Gives information for the process identification.</td>
</tr>
<tr>
<td>Due Date</td>
<td>Date</td>
<td>Event</td>
<td><code>duedate</code></td>
<td>Gives information about delays in activities execution.</td>
</tr>
<tr>
<td>Creation Date</td>
<td>Date</td>
<td>Event</td>
<td><code>time : timestamp</code></td>
<td>Gives information for the execution time identification.</td>
</tr>
<tr>
<td>Handled Date</td>
<td>Date</td>
<td>Event</td>
<td><code>time : timestamp</code></td>
<td>Gives information for the execution time identification.</td>
</tr>
<tr>
<td>Dossier Id</td>
<td>String</td>
<td>Trace</td>
<td><code>concept : name</code></td>
<td>Gives information for the trace identification.</td>
</tr>
<tr>
<td>Activity</td>
<td>String</td>
<td>Event</td>
<td><code>concept : name</code></td>
<td>Gives information for the activity identification.</td>
</tr>
<tr>
<td>NA</td>
<td>String</td>
<td>Event</td>
<td><code>lifecycle : transition</code></td>
<td>Gives information about the status of an activity, and it's obtained by the detection of the creation Date or the Handled Date.</td>
</tr>
</tbody>
</table>

Table 3.1: Process Order log definition.
CHAPTER 3. PRELIMINARY ANALYSIS

3.1.3 Process Analysis

In order to validate the current approach selected in this research, we made a short analysis of the structure of the process Order. The purpose of this is to analyze a process that is executed under a Case Handling (CH) conception (e.g. DossierFlow). We do this using the ProM 5 Heuristic Miner plug-in, which is able to retrieve a process structure. We illustrate this in Figure 3.3, where we can see that indeed it is not possible to use the process model obtained from the event log for a performance analysis. Due to the fact, that the structure is not readable and also that the sequence of activities is highly variable. Additionally, we used the standard parameters of the Heuristic Miner plug-in.

![Figure 3.2: Order's trace XES example.](image)

![Figure 3.3: Order process model](image)

In the process description we saw that the sales department of the Dutch company defined five activities for the process execution, which is contradictory with the process model showed above that has more than 20 activities. This situation happened because DossierFlow allows us to define
new activities during the process execution, which opens the door to the appearance of different labels for the activities. If the system users doesn't follow the official process definition.

3.1.4 Research Questions

In the previous chapter we saw that process performance is based on the detection of basic metrics, e.g. flow times, waiting times, processing times, resource utilization, etc. Additionally, we saw that those metrics are useful for validating process goals and how resources are dealing with them, e.g. resources versus delayed cases. This information can be extracted from an even log by the use of BI analytics such as key performance indicators (KPIs). These KPIs are based on specific process performance questions that once answered give the opportunity to the process owner to detect the current status of his process.

The sales department from the Dutch company wants to gather information about the performance of the process Order, for that they have defined the following questions:

1) Which is the percentage of resources' participation?

2) Which is the percentage of activities' participation?

3) How many cases were created and closed per month in 2010?

4) Which are the average, standard deviation and maximum values of the cases' durations per year?

5) How predictable is the working time spent in “Behandelen” in 2011 considering the average, standard deviation and maximum values?

6) Do I have cases in 2011 that had an average time higher than seven days between the completion of “Behandelen” and “Order kan ingevoerd worden”?

7) How many cases in 2010 and 2011 had activities that didn't accomplish their predefined “due date”?

8) Which resource had the highest participation as a main handler of the cases in question 7?

3.2 Case Study 2: Dutch Municipality

The second case study is about a Dutch municipality, which is in charge of processes that handle taxes, environmental permit applications, event organizing approvals, etc. Furthermore, this DossierFlow customer has requested an analysis of the process “Evenementenvergunningen” (Event Permits).

Additionally, we received the DossierFlow database from this customer, containing information about the Event Permits execution during the time period of May 2010 to March 2011.
3.2.1 Process Description

The process of Event Permits is related to registering and processing requests for organizing public events in this Dutch municipality. Furthermore, this process is focused on documents rather than process activities. This differs from the behavior that we saw in the previous case study, because in this case we need to associate the type of the documents to stages in the process.

For example, the process has a document type labeled “Aanvraag vergunning”, which is used for registering all the documents that are necessary to start processing the request. Furthermore, we can associate this type to the initial stage of the process, where the start and end of the stage is given by the first and last appearance of a document with the type “Aanvraag vergunning”. This association of document types with stages is explained in more detail in Appendix B.

Additionally, the Dutch municipality has the following Event Request document types:

- **Aanvraag vergunning**: associated with the documents required for starting the Event Permits application procedure.
- **Ontvangstbevestiging**: associated with the documents required after the request for Event Permit is acknowledged by the municipality.
- **Aanvullende gegevens**: associated with extra documents required in the analysis of the request.
- **Vergunning**: associated with the documents required after the municipality has decided to accept the request.
- **Brief afwijzing**: associated with the documents required after the municipality has decided to reject the request.

3.2.2 Information Structure

In the previous case study we realized that to be able to perform the Process Mining analysis with ProM we need to extract the information from DossierFlow database tables and construct an event log using XES. Plus, DossierFlow handles the information related to the process, documents and resources involved in its execution in three database tables. The fields selection criteria and the data type associated to them are shown in Table 3.2. (please continue to the next page)
Additionally, in Figure 3.4 we can see an example of a trace from the process Event Request using the XES format. (Please continue to the next page)
CHAPTER 3. PRELIMINARY ANALYSIS

3.2.3 Process Analysis

Following the procedure done in section 3.1.3, we make a short analysis of the structure of the process Event Permits. We illustrate this in Figure 3.5, where we can see again a process model that is not readable and that also has a highly variable sequence of activities. Additionally, we also used the standard parameters of the Heuristic Miner.

In the process model showed in the figure we can also see that the number of process activities or document types is also higher than the five types defined in process description. In this case the total number of document types was 15. This situation also happened because DossierFlow users can define extra document types during the execution of the process.(please continue to the next page)
Figure 3.5: Event Permits process model


3.2.4 Research Questions

Following the procedure done in section 3.1.4. We identified the following process performance questions that the Dutch municipality wants to respond about the process Event Permit:

1) Which is the percentage of resources' participation?

2) Which is the percentage of document types' participation?

3) How many cases entered per month in 2010?

4) Which are the average, standard deviation and maximum values of the cases' duration in 2010? (Note: for the municipality the duration of the case is based on the time between the completion of “Aanvraag vergunning” and the completion of “Vergunning”).

5) Which are the average, standard deviation and maximum values of the time between the completion of “Ontvangstbevestiging” and the completion of “Vergunning”, in 2010?

6) Considering question 5, which cases had a value higher than the average + the standard deviation?

3.3 BPA Critical Evaluation

After introducing the case studies on the previous section with their process performance questions, we perform a critical evaluation of the Basic Performance Analysis BPA. This evaluation is done by using the BPA for answering the questions of each case study.

3.3.1 BPA description

The Basic Performance Analysis\(^4\) (Figure 3.6) is a plug-in for ProM 5 that calculates performance measures like execution times, waiting times, etc. It has the potential of displaying performance metrics using several types of graphs such as Bar Charts, Pie Charts, etc.

Regarding the way that the BPA analyzes the information, we can distinguish three basic dimensions:

- **Instance**: dimension related to the case perspective of the process.
- **Task**: dimension related to the activity perspective of the process.
- **Originator**: dimension related to the resource perspective of the process.

\(^4\) See [http://www.processmining.org/online/basicperformanceanalysis](http://www.processmining.org/online/basicperformanceanalysis)
CHAPTER 3. PRELIMINARY ANALYSIS

With these dimensions, it is possible to have any kind of combinations like: Instance vs. Task, Task vs. Originator, etc. Additionally, we can consider as well, a Time dimension for tracking the behavior of one of the basic dimensions mentioned before. Once a desired dimension is selected, it is possible to have different types of performance measures like:

- **Working time**: the time involved for executing a particular activity.
- **Waiting time**: the time involved between the assignment of an activity and the starting of it.
- **Sojourn time**: the time between the beginning of a case and the last activity registered of it.
- **Frequency**: the number of times a particular activity, resource or case occurred in relation to the Working time or Waiting time captured.

Finally, it is possible to have different types of metrics related with the performance measures mentioned before, these types of metrics are: average, maximum, minimum, standard deviation and median.

### 3.3.2 Case study 1 evaluation

In the introduction of the case study 1, which is related to the sales department of a Dutch company; we were able to gather some performance questions of the process Order. We use the BPA for answering these questions.

**Question 1**: Which is the percentage of resources' participation?

![Figure 3.6: Basic Performance Analysis.](image-url)
This question cannot be answered directly with the BPA, because this plug-in gives a frequency based on the *Working Time or Waiting Time* measure.

**Question 2:** *Which is the percentage of activities' participation?*

This question cannot be answered directly as well with the BPA, because of the same reason explained in *Question 1.*

**Question 3:** *How many cases were created and closed per month in 2010?*

To answer this question it's necessary to track the frequency of the activities: “dossier gecreëerd” (case creation) and “dossier afgesloten” (case closed). Those are system activities logged in by default in DossierFlow when the case is created or closed. Additionally, the BPA cannot answer this question directly because it only tracks the time spent in an activity in a predefined period of time (e.g. not parameterizable).

**Question 4:** *Which are the average, standard deviation and maximum values of the cases' duration per year?*

The BPA cannot answer this question because it only makes a sum of the time spent per case and per month, which is shown in Figure 3.7.

![Figure 3.7: Order cases accumulated duration in hours per month.](image)

**Question 5:** *How predictable is the working time spent in “Behandelen” in 2011 considering the average, standard deviation and maximum values?*
For this question we make an assumption, we consider that an activity is predictable when it has a low standard deviation and a not so high maximum value, both in comparison with the average value. Additionally, the BPA is able to answer this question. The metrics related with the average, standard deviation and maximum values are shown in Figure 3.8, Figure 3.9, and Figure 3.10. From the figures we can conclude that “Behandelen” is not predictable, because it has a too high standard deviation value of 12.5 days and a too high maximum value of 90 days.
CHAPTER 3. PRELIMINARY ANALYSIS

**Question 6:** Do I have cases in 2011 that had an average time higher than seven days between the completion of “Behandelen” and “Order kan ingevoerd worden”?

The BPA cannot answer this question because it has predefined views and measures, and none of its options are able to track the time between the completions of two activities in a case.

**Question 7:** How many cases in 2010 and 2011 had activities that didn't accomplish their predefined due date?

The BPA doesn't use extra meta attributes of an event for calculations. Finally, we cannot answer this question.

**Question 8:** Which resource had the highest participation as a main handler of the cases in question 7?

We cannot answer this question because we need the answer of question 7.

### 3.3.3 Case Study 2 evaluation

In the introduction of the case study 2 that is related to the Dutch municipality, we were able to gather process performance questions that required an answer from the process Event Permits. We use the BPA for this purpose.

**Question 1:** Which is the percentage of resources' participation?

This question cannot be answered directly with the BPA, because this plug-in gives a frequency based on the Working Time or Waiting Time measure.

![Figure 3.10: Maximum working time of Behandelen.](image)
CHAPTER 3. PRELIMINARY ANALYSIS

**Question 2:** Which is the percentage of document types' participation?

This question cannot be answered directly as well with the BPA, because this plug-in gives a frequency based on the *Working Time or Waiting Time* measure.

**Question 3:** How many cases entered per month in 2010?

We cannot answer this question because we need to track the frequency of complete “*Aanvraag Vergunning*” per month. Furthermore, the BPA only retrieves values based on the *Working Time* per month and also doesn’t consider the use of filters for the event type and a year.

**Question 4:** Which are the average, standard deviation and maximum of the cases' duration in 2010? (Note: for the municipality the duration of the case is based on the time between the completion of “*Aanvraag vergunning*” and the completion of “*Verguning*”).

The BPA cannot answer this question because it has predefined views and measures, and none of its options are able to track the time between the completions of two activities in a case.

**Question 5:** Which are the average, standard deviation and maximum values of the time between the completion of “*Ontvangstbevestiging*” and the completion of “*Verguning*”, in 2010?

As explained in question 4, the BPA cannot track this metric.

**Question 6:** Considering question 5, which cases had a value higher than the average + the standard deviation?

We cannot answer this question because we need the answer of question 5.

### 3.4 Usability

Once we have analyzed the BPA for answering process performance questions, we also perform an evaluation of its *usability* considering a heuristic approach [24] based on the following aspects:

- **Visibility of system status:** the system should always keep users informed about the system status.

- **User control and freedom:** users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue.

- **Consistency and standards:** users should not have to wonder whether different words, situations, or actions mean the same thing. Additionally, the system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
CHAPTER 3. PRELIMINARY ANALYSIS

- **Error prevention:** even better than good error messages is a careful design that prevents a problem from occurring in the first place.

- **Recognition rather than recall:** minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.

- **Error recognition:** error messages should be expressed in plain language (e.g. no codes), precisely indicate the problem, and constructively suggest a solution.

Additionally, we use a Low/Medium/High scale to evaluate the aspects mentioned above. The BPA usability evaluation is illustrated in Table 3.3, where we can see the values obtained by the BPA with an explanatory observation. In summary, we can see that the BPA got a low value for the usability criteria.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility of system status</td>
<td>X</td>
<td></td>
<td></td>
<td>The BPA only gives information about the system status when the plug-in calculates values for the first time and not when it requires extra calculations like in the Time Chart.</td>
</tr>
<tr>
<td>Consistency and standards</td>
<td></td>
<td>X</td>
<td></td>
<td>Sometimes the labels get mixed, e.g. “Measure” is used for Statistical functions and for retrieving the throughput of a process instance. On the other hand the chart labels are configured automatically. e.g. labels of axis.</td>
</tr>
<tr>
<td>User control and freedom</td>
<td>X</td>
<td></td>
<td></td>
<td>Once a selection is made there is not possible to stop the calculation.</td>
</tr>
<tr>
<td>Error prevention</td>
<td>X</td>
<td></td>
<td></td>
<td>The system doesn't give alerts about possible errors, e.g. we can insert numeric values in text fields.</td>
</tr>
<tr>
<td>Recognition rather than recall</td>
<td></td>
<td>X</td>
<td></td>
<td>Most of the calculations are done at the beginning, which reduces the interaction with the system for processing most of the charts. This with some exceptions like in the Time Chart and Meter Chart.</td>
</tr>
</tbody>
</table>

Table 3.3: BPA usability evaluation

3.5 Findings

In this preliminary analysis we introduced the two case studies used in this research, which are current DossierFlow customers. The first case is related with a process for handling orders in a sales department of a Dutch company and the second is related with a procedure for processing public
event organizing requests of a Dutch municipality. Furthermore, we could detect that both of them describe the process with statuses or stages. In the case of the sales department they define these statuses, but in the case of the Dutch municipality they just use document types. Additionally, we used a different approach for the municipality that associates these document types with the stages of the process. Finally, the behavior that both case studies presented can be associated as well to a Case Handling (CH) orientation, which was explained in the previous chapter and is focused on what has to be done to finish the case [8].

After describing the case studies we gathered process performance questions defined by the process owners. Furthermore, we used these questions for validating the BPA plug-in from ProM 5. Additionally, we could see that the BPA was not able to give answers to most of the questions, because it has predefined dimension views and measures that are limited for retrieving specific process performance information as the ones required by both case studies. For example, it bases the whole performance analysis in the Working time measure, which limits the possibility of seeing basic information, such as how many process cases entered per month. Also, it doesn't allow us to track the time between the completions of two different activities in a case execution, which is useful for determining process service times.

Regarding visualization capabilities, the current version of the BPA doesn't allow us to control the information displayed on the chart axes. Furthermore, it is limited on grouping capabilities. For example it is not possible to decide the time unit, e.g. instead of months we can select weeks. Additionally, we can't adapt it for new visualizations, e.g. comparing the working time of 2010 vs. 2011 in a single plot. This lack of adaption is related to its predefined dimensions, because it only considers cases, resources and activities. Additionally, the BPA is attached to a MXML format that is limited for including new meta-data elements with data type in the event attributes definition, e.g. Cost with the type Real.

About the tool performance, the BPA requires filtering capabilities for the calculation and visualization, because the BPA starts calculating working times for the entire log without determining any type of boundary for it. This can be time consuming when we only want to analyze a small part of the log, and also in some cases it produced an "out of memory error". We also saw that some charts become unreadable, for example when more than 20 resources appear on the same axis. Additionally, the BPA has a lack in reporting capabilities, which are required by Océ customers in order to transmit the results obtained from the performance analysis.

Finally, we saw that the BPA's usability was poor against some criterion such as: visibility of system status, user control and freedom, error prevention and recognition rather than recall. Additionally, after analyzing the current version of the BPA, we conclude that there are severe limitations in terms of filtering, selection and visualization. Furthermore, this supports the solution proposed in this master thesis project of implementing a new plug-in that will allow us to do a Generic Process Performance Analysis (GPPA), which must not have dimension limitations and also must be able to filter, group and visualize the event log information in a flexible and configurable way.
Chapter 4

Meta-model definition

This chapter presents the meta-model definition (please follow Figure 4.1) for graphical representation of properties of event logs, which is composed by different elements such as Events, Charts, Axes, Groups, Mappings and more. This allows us to have flexibility in the process performance analysis and also to generate attached data visualizations, e.g. the time between two activities per year. Additionally, in Section 4.1 we define the meta-model, in Section 4.2 we use the meta-model for defining the mappings that will be implemented for the GPPA, in Section 4.3 we discuss the performance chart visualization and finally in Section 4.4 we define the structure that will be used for the performance report generation.

Figure 4.1: Research Goals – Meta-model Definition.
4.1 Meta Model Definition

The UML (Unified Model Language) class diagram shown in Figure 4.2 describes the meta-model that we use for graphical representation of properties of event logs. In the following, the meta-model components are introduced in more detail.

![Figure 4.2: Meta-model for graphical representation of properties of logs.](image-url)
4.1.1 Log

On the top level there is one log object, which contains all the event information that is related to one specific process (e.g. “Orders”, “Event Permits”) [22].

4.1.2 Trace

A log contains an arbitrary number of trace objects. Each trace describes the execution of one specific instance, or case, of the logged process [22]. For example, one specific Order or Event permit processed.

4.1.3 Event

Every trace contains an arbitrary number of event objects. Events represent atomic granules of activity that have been observed during the execution of a process [22]. As such, an event has no duration. For example, recording the information related to an event permit request.

4.1.4 Property

The log, trace, and event objects contain no information themselves. All information in an event log is associated with properties that describe their parent element. Additionally, the properties have a type (String, Integer, Real, Boolean, Date), which is related to the type of information (value) that they contain. For example, one property of an event could be the name of the activity that is referred to. We illustrate this using the notation of XES in Figure 4.3 to describe one event of the process Event Permits. In the figure we can see the relationship between an event with the property concept: name. Furthermore, we also can see the type of value that concept: name has which is String.

```
<event>
  <string key="concept:name" value="Aanvraag vergunning"/>
  <string key="org:resource" value="John"/>
  <date key="time:timestamp" value="2009-07-13T09:22:54.000+02:00"/>
  <string key="lifecycle:transition" value="start"/>
</event>
```

Figure 4.3: XES event property example of the process Event Permits.
4.1.5 Group

We define a group as a set of events that is constructed using a particular grouping criteria. Additionally, a group is identified by a group name and it can have an arbitrary number of metrics. These are obtained by calculations over the property values of the events that the group contains. For example in XES, we can group events according to their concept: name property value. Furthermore, we can count the number of events that this group has, and use it as a group metric. This is illustrated in Figure 4.4, which shows the construction of a group with these characteristics from two process instances of Event Permits.

![Figure 4.4: Group construction using the concept:name property](image)

4.1.6 Mapping

Mappings are functions that allow the creation, filtering and metric assignment of groups. This, using a mapping method, which requires an arbitrary number of setting parameters for the mapping method configuration. For example, in Figure 4.4 we can say that we used a mapping for the creation of the group. This mapping has one parameter for grouping the events that is the concept: name property value of each one of them.

Additionally, we use the following notation for a mapping:

\[ \text{mappingName (parameter 1,parameter 2,parameter 3...parameter n)} \]
We have considered three main classes of mappings: grouping, filter and measurement.

4.1.6.1 Grouping
This class of mapping generates a group or groups of events using an arbitrary number of parameter values. e.g. “Group events using their concept: name property value”. The use of a grouping was illustrated in the example shown in Figure 4.4.

4.1.6.2 Filter
This class of mapping filters events in a group using an arbitrary number of parameter values. For example, we can apply a filter to our previous grouping example (Figure 4.4) by adding a filter mapping that only considers events where the org: resource property value is equal to “John”. This is illustrated in Figure 4.5, which shows a reduction in the previous group metric value, because now there is only one event that complies with the filtering criteria.

![Diagram showing filter mapping example using the org:resource parameter value “John”]

**Figure 4.5**: Filter mapping example using the org:resource parameter value “John”

4.1.6.3 Measurement
This class of mapping is used for generating a metric or measure to a group of events. This is accomplished, by using an arbitrary number of parameter values. For example, in our previous
grouping example shown in Figure 4.4, we can say that we used a measurement mapping for calculating the number of events that a group has.

4.1.7 Axis

The axis is a line used as a reference for drawing on a graph, which has a label or name, and a type that can be continuous or discrete. This type distinction is related to the type of information that is mapped onto the axis. We consider that discrete values are String, Integer and Boolean values. Additionally, we consider that continuous values are: Real and Date values.

4.1.8 Chart

It's a graph used for making a graphical representation of data, which is composed of axes (continuous or discrete) that are used for showing a group, or groups' information. For example, we can use a chart for displaying the groups' names onto a discrete axis and the metrics associated with them onto the continuous one. Additionally, a chart can be classified according to a type of chart (e.g. Bar Chart). This would be, according to the number of axes that a chart contains, and to the types of its axes, e.g. continuous or discrete.

For example, we consider that a Pie Chart is a type of chart that has two axes, the first one is discrete and contains the names of the groups (name of each slice) and the second one is continuous and contains the values associated to each group (relative size of each slice). This is illustrated in Figure 4.6, which presents a Pie Chart that shows the number of events (group metric) per activity (group name).

![Activities Frequency](image)

**Figure 4.6:** Number of events per activity
4.1.8 Plot

We define a plot as a combination of a chart with groups (with mappings) used for the graphical representation of properties of logs. Additionally we use the following notation for a plot:

\[ P=(C,G,F,V) \]

Having that \( P \) is a plot, \( C \) a type of chart, \( G \) is a set of grouping mappings \( (g_1, g_2, \ldots, g_n) \), \( F \) is a set of filter mappings \( (f_1, f_2, \ldots, f_n) \) and \( V \) is a set of measurement mappings \( (v_1, v_2, \ldots, v_n) \). In addition, \( G, F, V \) describe the mappings used for the groups construction, filtering and metric assignment.

For example, we define a plot for the example used in Figure 4.6:

\[ P=(C=\text{Pie Chart}, G=(\text{groupByProperty(concept:name)}), F=\text{noFilter}, V=(\text{countEvents()})) \]

In this definition we can see that a Pie Chart is plotted using a grouping mapping that we call groupByProperty, which constructs groups of events according to their concept: name property value. Additionally, we don't use a filter (e.g. noFilter) and we use a measurement mapping that we call countEvents for counting the number of events for each group.

4.2 Mappings Library

We have defined a set of predefined mappings that will be implemented later on (next chapter) for the GPPA. Additionally, we have classified them according to the three main classes for applying mappings to the log information: grouping, filter and measurement.

4.2.1 Grouping Mappings Library

We have considered five grouping mappings: Group General, Group Numeric, Group by Period, Group by Time Unit and Group by Time Unit Static.

4.2.1.1 Group General

These classes of mapping generates groups of events in accordance to a discrete parameter value (Boolean, String and Integer). For example, in our grouping example of Figure 4.4 we grouped events using the events’ concept: name property value that is a String.

We define a Group General mapping using this notation:

\[ \text{GroupGeneral}\{\text{Property}\} \]

Having that Property is an event log property.
4.2.1.2 Group Numeric

This class of mapping generates groups of events using an *Integer* or *Real* parameter value. This is accomplished by defining intervals (e.g. maximum and minimum values.) for each one of the desired groups.

For example, Figure 4.7 shows a *Bar Chart* that has names of groups of events constructed using a parameter age, onto the discrete axis. This parameter is a self-defined *Integer* property that we use for describing events. Additionally, we create the groups using age intervals for each group, e.g. “group 1 to 10 is related to events that have an age property value that is between 1 and 10”. Once we have the groups of events, we calculate the frequency (number of events) of each one of them and map these values onto the continuous axis.

![Frequency per age](image)

**Figure 4.7:** Frequency of resources per age intervals.

We use the following notation for a Group Numeric mapping:

\[
\text{GroupNumeric}(\text{Property}, \text{MinValue}, \text{MaxValue})
\]

*Property* is considered an event log property, and *MinValue* and *MaxValue* are defined as interval boundaries.

4.2.1.3 Group by Period

These mapping generates groups of events that occurred in different periods of times using their *Date* parameter value. It is composed of a “From” (Date) - “To” (Date) structure, which will allow
us to define time boundaries for each one of the periods of times selected for the group creation. For example, Figure 4.8 shows a Bar Chart that has periods of times (groups’ names) placed onto the discrete axis, and the frequency of activities onto the continuous axis (groups’ metrics) associated with them.

![Bar Chart](image)

**Figure 4.8**: Frequency of Activities per time period.

We use the following notation for a *Group by Period* mapping:

\[ \text{GroupDate(Property, From, To)} \]

Having that *Property* is an event log property, and that *From* and *To* are defined as period boundaries.

### 4.2.1.4 Group by Time Unit

This mapping also groups events in periods of times using a *Date* parameter value. The difference with the *Group by Period* mapping definition is that we automatically create a set of groups. First we detect the time value of the earliest timestamp of the event log, which is related to the difference expressed in milliseconds between this timestamp and a predefined reference *Date* that is the 1970-01-01. Once we have this time value, we add a time unit value (year, month, week, day, hour, minute and second) to it until we reach the latest timestamp of the event log.

Additionally, the boundaries of each group created are defined each time that a time unit is added.
CHAPTER 4. META-MODEL DEFINITION

For example, if the earliest timestamp is the “2011-02-01 00:00:00” and we decided to group events by day. The first period of time will go from the “2011-02-01 00:00:00” until the “2011-02-01 00:00:00” + a day value (24 \cdot 60 \cdot 60 \cdot 1000 = 86400 milliseconds). Hence the To period of time value will be “2011-02-02 00:00:00”. Finally, once we have defined a period of time, we can detect the events that were executed in that period. This can be achieved by using an event Date property value, which is related to the Date parameter selected for the mapping configuration.

We use the following notation for a Group by Time Unit mapping:

\[ \text{GroupByTu}(\text{Property}, \text{TimeUnit}) \]

Having that Property is an event log property, and TimeUnit is the time unit selected for the groups’ construction.

### 4.2.1.4 Group by Time Unit Static

This mapping groups events using a Date parameter value and a static time unit (Years, Months of the year, Weeks of the year, Days of the week, Hours of the day, Minutes of an hour, and Seconds of a minute). Furthermore, this mapping also creates a set of groups automatically. First we create a set of groups according to the static time unit value selected. For example, if we chose “Months of the year”, 12 groups will be generated for each month of the year.

Additionally, once we have defined the groups, we analyze the static time unit value of an event Date property value (related to the Date parameter) to determine if that event belongs to a particular group. For example, if we have “2011-02-02 00:00:00” as Date property value, then we can say that this event belongs to the group “February”. To illustrate this, in Figure 4.9 we show a Pie Chart that shows the frequency of events per month of the year.

We use the following notation for a Group by Time Unit Static mapping:

\[ \text{GroupByTUS}(\text{Property}, \text{TimeUnitStatic}) \]

Property is considered an event log property, and TimeUnitStatic is defined as the time unit static selected for the groups’ construction. (please continue to the next page)
### 4.2.2 Filtering Mappings Library

We have considered four filtering mappings: General filter, Numeric filter, Date filter and Date Comparator.

#### 4.2.2.1 General Filter

It is used for filtering events in a group using a discrete (String, Boolean and Integer) event property value, e.g. “selecting the match activities for the inclusion of an event” or “selecting events that have a ‘true’ value on its Boolean property”. This is useful for handling logs with big amounts of information, which might involve high processing times and complex visualizations. Furthermore, this filter allows us to select the part of information of the log that we want to see.

Additionally, as an extension of this filter, we also have considered an extra option for filtering a group of events as a whole. For example, if we grouped events using the Group Numeric mapping, we can select the intervals that we wanted to see in the plot chart. This can be achieved by using the attribute name (String) value that each group has.

Finally, the filtering extension only applies for groups of events that were not created automatically. Hence Group by Time Unit and Group By Time Unit Static are not considered.

We use the following notation for a General filter mapping:

\[
\text{General}(\text{Property}, \text{TargetValue})
\]

*Property* is considered an event log property or a group name, and *TargetValue* is defined as the set of desired values.

---

**Figure 4.9:** Frequency of events per month of the year

<table>
<thead>
<tr>
<th>Month</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>444</td>
<td>7%</td>
</tr>
<tr>
<td>Feb</td>
<td>806</td>
<td>13%</td>
</tr>
<tr>
<td>Mar</td>
<td>545</td>
<td>9%</td>
</tr>
<tr>
<td>Apr</td>
<td>604</td>
<td>10%</td>
</tr>
<tr>
<td>May</td>
<td>624</td>
<td>10%</td>
</tr>
<tr>
<td>Jun</td>
<td>501</td>
<td>8%</td>
</tr>
<tr>
<td>Jul</td>
<td>420</td>
<td>7%</td>
</tr>
<tr>
<td>Aug</td>
<td>379</td>
<td>6%</td>
</tr>
<tr>
<td>Sep</td>
<td>501</td>
<td>8%</td>
</tr>
<tr>
<td>Oct</td>
<td>327</td>
<td>5%</td>
</tr>
<tr>
<td>Nov</td>
<td>402</td>
<td>7%</td>
</tr>
<tr>
<td>Dec</td>
<td>427</td>
<td>7%</td>
</tr>
</tbody>
</table>
4.2.2.2 Numeric Filter

It is used for filtering events in a group using a Real or Integer event property value, e.g. “Selecting the events that have a self-defined cost property value higher than one hundred”. Additionally, just as we proceeded with the General Filter, we also have an extension of this mapping for filtering a group of events as a whole. This can be achieved by using one of the metrics associated to a group. For example, if we have a metric that counts the number of events that a group contains, we can have a filter that only considers the groups that have more than one hundred events.

Additionally, we use these value comparators: >, <, ≥, ≤, ≠. and the following notation for a Numeric filter mapping:

\[
\text{NumericF}(\text{Property}, \text{TargetValue}, \text{ValueC})
\]

Property is considered an event log property or a group metric value, TargetValue is defined as the desired value, and ValueC is known as a value comparator.

4.2.2.3 Date Filter

The Date filter is useful for determining boundaries for the log information extraction. This mapping uses an event Date property value for filtering events in a group, e.g. “Selecting the events that have a timestamp between the 1st of January and the 30th of July of 2011”. Additionally, we need a “From” Date and a “To” Date, to determine the period of time that we will use as criteria for an event selection.

We use the following notation for a Date Filter mapping:

\[
\text{DateFilter}(\text{Property}, \text{From}, \text{To})
\]

Property is considered an event log property, and From and To are Date values used for defining the boundaries of the filter.

4.2.2.4 Date Comparator

In the preliminary analysis we realized that in the case study about the sales department of a Dutch company, they need to detect when an activity didn't accomplish its predefined due date. Hence we need a mapping that compares two event Date property values. For that, we use the Date Comparator mapping, which makes this Date properties values comparison for determining if a particular event must be included in a group.

Additionally, we use these Date value comparators: >, <, ≥, ≤, ≠. and the following notation for a Date Comparator mapping:

\[
\text{DateComparator}(\text{Pr1}, \text{Dc}, \text{Pr2})
\]

Pr1 and Pr2 are considered event log properties, and Dc is known as a Date value comparator.
4.2.3 Measurement mappings

We have considered four measurement mappings: General Measurement, Time Between, Activity Working Time and Activity Working Time per Group.

4.2.3.1 General Measurement

This mapping retrieves a numerical value from a group of events and assigns that value to a group's metric. It uses an event discrete (Integer, Real and String) parameter value of an event. In the case of Integer or Real values, it takes the property value and uses it as a statistical reference. This is done with the goal of retrieving values such as Average, Frequency, Standard Deviation (STD), Median, Sum, Max, and Min. In the case of a String parameter value, we don't take the property value directly for a calculation and just use it as reference for counting the number of occurrences that it has in a group, e.g. “Frequency of activities”.

We use the following notation for a General Measurement mapping:

\[ \text{GMS(Property,StatisticalF)} \]

Property is considered an event log property and StatisticalF is known as the statistical function that we want to use for retrieving the metric.

4.2.3.2 Time Between

This mapping is used to determine the elapsed time between the executions of two activities in a case or process instance. For example, in the first case study we realized that it is required to know if the elapsed time between “Behandelen” and “Order kan ingevoerd worden” was higher than seven days. This mapping is pre-configured for a event property activity that in XES is concept: name. Additionally, it only requires the names of the activities (target values) for determining the time between them.

The calculation process starts with the identification of traces in a constructed group and the temporary creation of sub groups of events according to the trace that they belong. Then we analyze the activity property value of each event versus our activities target values, if we find the first target value we keep the timestamp property of that event. A similar procedure is performed for the second activity target value. Once we have identified both timestamps we calculate the difference (expressed in milliseconds) between them and we keep that value for the group metric calculation. In the case that one of the activity target values was not detected inside a subgroup, we consider a value of zero for that particular subgroup.

Finally, we have considered as well for retrieving a group metric, statistical functions such as Average, Frequency, Standard Deviation (STD), Median, Sum, Max, and Min. This in combination with a granularity option for expressing the results in time units such as year, month, week, day, hour, minute and second.

We use the following notation for a Time Between mapping:
CHAPTER 4. META-MODEL DEFINITION

\[ TimeBetween(Tg1, Tg2, StatisticalF, GranularityV) \]

\( Tg1 \) is the name of the first activity (e.g. Behandelen) and \( Tg2 \) is the name of the second activity (e.g. Order kan ingevoerd worden). Additionally, we have \( StatisticalF \) for the statistical function and \( GranularityV \) for the granularity value.

4.2.3.3 Activity Working Time

The Activity Working Time (AWT) is a mapping that allows us to determine the time spent in the execution of a process activity. Furthermore, we can see this time under different perspectives according to the event properties available in a log. For example, we can see the AWT per activity or per resource. We have considered as well, working schedules for the time estimation, e.g. a company works from 8:00 to 16:00 and not on weekends.

Additionally, the AWT considers the activity Life Cycle status for the time calculation, because it measures the time between (considering the event timestamp) the beginning and the completion of a particular activity. It also requires to group events per trace from the whole log in order to have a more realistic metric that reflects the time occurred per process instance for a particular activity. This implies that we cannot use the same approach applied for the previous mappings because the AWT retrieves a measure that is independent of a previous grouping definition. For example if we grouped events per resource, we cannot just read each one of the events per group, because we need to discompose the groups and generate groups of events according to the traces that they belong to.

Hence, we use a different approach for the AWT calculation that uses the previous group definition but only considering the group label or name for the metric assignation. For example, first we group events per activity and then the AWT will retrieve a metric for each group without considering the events contained by them and only will take in account the name that they have. Additionally, the AWT can only retrieve metrics for groups that have a name based on event String property, e.g. concept: name, org: resource, etc.

Additionally, we made the following assumptions for the time estimation:

- If an activity only has “complete” as value for Life Cycle status, we use the closest timestamp to determine where it begins. This is achieved by comparing the timestamp of the previous event executed in the trace, with the timestamp of the closest event of the previous trace where the resource involved in it, is the same one that completed the activity.

- The metric is assigned to an event property value when the activity has been completed. For example, if we selected resource for the AWT calculation, we assume that the resource that was involved in the completion of the activity is the one that started it.

Finally, we have considered as well, retrieving the AWT metric, with the use of statistical functions such as Average, Frequency, Standard Deviation (STD), Median, Sum, Max, and Min. This in combination with a granularity option for expressing the results in time units such as year, month, week, day, hour, minute and second.
CHAPTER 4. META-MODEL DEFINITION

We use the following notation for an AWT mapping:

\[
AWT(Property, From, To, Weekends, StatisticalF, GranularityV)
\]

Having that Property is an event String property value, h1 and h2 define the start and end working hour, e.g. the company works from 8:00 to 17:00. And Weekends is a Boolean value for determining if weekends are considered on the calculation. Additionally, we have StatisticalF for the statistical function and GranularityV for the granularity value.

4.2.3.4 Activity Working Time per Group

The Activity Working Time per Group (AWTG) is an extension of the AWT that considers the events of a group and not the entire log for the activity duration calculation. We do that for calculating working time measures in groups constructed by time periods, where we need to see specifically parts of the log. Additionally, we use a similar notation for an AWTG mapping:

\[
AWTG(Property, From, To, Weekends, StatisticalF, GranularityV)
\]

4.3 Charts Visualization

After the explanation of the mappings library, the next step is to address its visualization on a chart. For that purpose, the GPPA offers three basic types of charts: Bar Chart, Pie Chart and Line Chart. These are used, because of their popularity in the market, their usefulness for data series comparison at the glance (Pie Chart & Bar Chart) and also because they are able to show trends (Line Chart) [25], [26].

4.3.1 Bar Chart

Bar Chart is a type of graph used for displaying and comparing numerical values, e.g. “Frequency of resources”. It is one of the most commonly used types of graph, because they are easy to create and easy to interpret [27]. Additionally, it is a flexible chart able to be displayed in different variations as: horizontal bar charts, grouped or component charts (series, multiple continuous axes). Furthermore, the Bar Chart in the context of the GPPA, considers three presentation formats: Simple Bar Chart, Bar Chart with Series and Bar Chart with more than one continuous axis.

4.3.1.1 Simple Bar Chart

A Simple Bar Chart has two axes; the first one is discrete and has the name of each bar. The second axis is continuous and has numeric values represented on the height of the bars.

4.3.1.2 Bar Chart with Series

A Bar Chart with Series has three axes; the first is discrete and has the names and colors associated with the series. The second is also discrete and has the name of each bar. The third is continuous and has numeric values represented on the size of the bars.
4.3.1.3 Bar Chart with extra continuous axes

A Bar Chart with extra continuous axes has $n$ axes; the first one is discrete and has the names and colors associated with the continuous axes. The second one is also discrete and has the name of each bar. The rest of the axes are continuous and have numeric values represented on the height of the bars.

4.3.2 Pie Chart

The Pie Chart is a circular graph that shows the relative contribution that different categories contribute to an overall total. A wedge of the circle represents each category’s participation, where 1% is related to a slice with an angle of 3.6 degrees [28].

Additionally, we define a Pie Chart in terms of axes:

A Pie Chart has two axes, the first one is discrete (the name of each slice) and the second one is continuous (the relative size of each slice).

4.3.3 Line Chart

The Line Chart handles similar elements as the Bar Chart; it only differs in the use of data points instead of bars, which makes it more effective for showing trends. Furthermore, the Line Chart in the context of the GPPA considers three presentation formats: Simple Line Chart, Line Chart with Series and Line Chart with more than Y axis.

4.3.3.1 Simple Line Chart

A Simple Line Chart has two axes; the first one is discrete and has the name of each line. The second axis is continuous and has numeric values represented on the line points.

4.3.3.2 Line Chart with Series

A Line Chart with Series has three axes; the first one is discrete and has the names and colors associated with the series. The second one is also discrete and has the name of each line. The third one is continuous and has numeric values represented on the line points.

4.3.3.3 Line Chart with extra continuous axes

A Line Chart with extra continuous axes has $n$ axes; the first one is discrete and has the names and colors associated with the continuous axes. The second one is also discrete and has the name of each line. The rest of the axes are continuous and have numeric values represented on line points.

Additionally, the Line Chart can be extended to a Time Series Chart if the ByTimeUnit mapping is used in the continuous axis. Like it is shown in Figure 4.10, where we can see the frequency (using
the General Measurement Mapping) of events per day (using the ByTimeUnit Mapping).

Figure 4.10: Line Chart extension to Time Series Chart.

4.4 Report Generation

Once the construction and visualization of the performance charts have been covered, the final step regarding the information presentation is the definition of the report structure. This step is performed in relation with the second problem statement, and is a common requirement from both case studies, which consists in giving reporting capabilities to the GPPA. The goal is that users of the GPPA must be able to generate a process performance report that is obtained in three steps. The first step is to generate the performance chart, the second is to add a description to the chart (this is done by GPPA user) and the third one is the report generation with a definition of a cover and summary (also done by GPPA user).

Additionally, for the report generation we consider a basic structure based on an example given by the Montfort University of Leicester [29], which is composed by: Cover and summary, Index section and Results.

4.4.1 Cover and Summary

One of the most-read parts of a report, are used to attract users' attention and to make a quick and short analysis of what has been developed in the report. Figure 4.11 shows the structure of this section that covers the Title, Author, Date, and Summary of the report.
4.4.2 Index Section

It is useful for determining the location of a topic in the report; Figure 4.12 shows an example of the format selected for it.

4.4.3 Results

This section covers the performance analysis done with the GPPA. For that, we selected three components for the charts explanation: title, description, and chart image. Those components we can see in Figure 4.13, where we present an example of this section.
4.5 Summary

In this chapter we defined the meta-model that we used to represent properties graphically in a log. We analyzed each element of this meta-model and the relationship between them. Furthermore, we focused on two elements in particular, which are mappings and charts. The first ones allowed us to create, filter and retrieve metrics of groups of events. This was achieved by using a criteria or calculation method. Additionally, we combined mappings with the use of charts to facilitate the visualization of process performance information. Finally at the end of the chapter, we saw the structure of the PDF report that we will use for presenting a consolidated process performance analysis.
Chapter 5

Solution Implementation

In the previous chapters we discussed the need for implementing a new plug-in (please follow Figure 5.1) that allows us to do a generic process performance analysis (GPPA). Furthermore, we did a preliminary analysis of two case studies. This analysis retrieved specific process performance questions required from DossierFlow customers. Additionally, we evaluated a previous plug-in solution (BPA from ProM 5) using these questions. The result of that analysis was that the BPA was not able to respond to the performance questions because it lacks flexibility for selecting and working with the attributes available in a event log.

Considering this flexibility required for answering process performance questions, the implementation of the GPPA plug-in was based on a chart representation approach that uses a meta-model to represent the event log properties graphically. These meta-model uses concepts such as data types with filtering, grouping and measurement mappings, which allow us to display in different perspectives the event log information, e.g. number of cases per user or per month of the year. Additionally, ProM 6 is the current implementation environment, which is supported by the log format XES. This format matches perfectly with the meta model definition, because we can use data types like String, Integer, Date, Boolean and Real. Furthermore, we implement the GPPA based on XES.

The chapter presents a solution overview in Section 5.1, the chart configuration and visualization in Section 5.2 and Section 5.3. Finally, we observe the report generation in Section 5.4 and in Section 5.5 we show the findings of this chapter. (please continue to the next page)
5.1 Solution proposal

For the implementation of the GPPA, a solution is proposed in Figure 5.2. First a log was defined in the preliminary analysis for each one of the case studies; this can be achieved by using the XES format. Furthermore, on ProM 6 we implement the GPPA to do the chart configuration and visualization. Finally, the solution ends with the GPPA report generation. (please continue to the next page)

Figure 5.1: Research Goals – Tool Implementation
5.2 Chart configuration

For the chart configuration we implement the mappings defined in the previous chapter. This is done by using configuration panels that allow us to select the parameters that are required for the mappings’ execution. Furthermore, the GPPA has a tab structure for distinguishing the first discrete axis (X axis) of the Pie Chart, Line Chart and Bar Chart. Additionally, we have tabs for the continuous axes (Y axes), Series and Filtering mappings definitions. This tab structure is illustrated in Figure 5.3, which shows each one of the components mentioned above.

Additionally, we have considered a building blocks approach for the mappings configuration, in which the GPPA user has to select step by step which parameters he/she wants to use before calling a particular mapping. This is illustrated in Figure 5.4, which shows that if we want to use the Group by Date mapping we have to first select an event Date property. Furthermore, we have implemented the buttons “Select More Mappings” and “Delete Mapping” for this mapping construction.

Finally, we show the relationship between the mappings definitions made in Chapter 4 with their configuration panels in the GPPA.

Note: Some of the names that appear on the configuration panels might be different than those in the mapping definition, because the labels have been modified in order to make them more user
friendly. Additionally, this doesn't mean that their functionality was modified.

5.2.1 Grouping Mappings Library

We have considered five grouping mappings: Group General, Group Numeric, Group by Period, Group by Time Unit and Group by Time Unit Static.
CHAPTER 5. TOOL IMPLEMENTATION

5.2.1.1 Group General

A Group General mapping uses the following notation:

$$GroupGeneral(Property)$$,

$Property$ is considered an event log property.

Additionally, we implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.5, which shows an example of the Group General panel that allows us to select org: resource as Property parameter from a list that contains the properties available in an event log.

![Figure 5.5: Group General (By Group) configuration panel.](image)

5.2.1.2 Group Numeric

A Group Numeric mapping uses the following notation:

$$GroupNumeric(Property, MinValue, MaxValue)$$,

$Property$ is considered an event log property, and $MinValue$ and $MaxValue$ are defined as interval boundaries.

We implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.6, which shows an example of the Group Numeric panel using Costs as Property parameter, “0” for the $MinValue$ and “100” for the $MaxValue$. Additionally, we have designed this panel to be able to create more than one group at the same time. This is supported by the buttons “Add” and “Delete” that allows us to create (“Add”) or delete groups created with this mapping. Finally, the “Add” button calls an extra configuration panel that allows us to select the $MinValue$ and $MaxValue$. 

Generic Process Performance Analysis 56
5.2.1.3 Group by Period

A Group by Period mapping uses the following notation:

\[ \text{GroupDate(Property, From, To)} \]

*Property* is considered an event log property, and *From* and *To* are defined as period boundaries.

We implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.7, which shows an example of the Group By Period mapping panel using timestamp as Property parameter, “2011-01-01 00:00:00” for From parameter and “2011-03-01 00:00:00” for To parameter. Additionally, we have designed this panel to be able to create more than one group at the same time. This is supported by the buttons “Add” and “Delete” that allows us to create or delete groups that have been created with this mapping. The “Add” button calls an extra configuration panel that allows us to select the From and To parameters. Finally, in the extra configuration panel we have a “Choose Date” button, which allows us to select a Date value from a calendar, e.g. “2011-03-01”.

![Figure 5.6: Group Numeric (By Interval) configuration panel.](image-url)
5.2.1.4 Group by Time Unit

A Group by Time Unit mapping uses the following notation:

\[ \text{GroupByTu}(\text{Property}, \text{TimeUnit}) \]

\(\text{Property}\) is considered an event log property, and \(\text{TimeUnit}\) is the time unit selected for the groups' construction.

We implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.8, which shows an example of the Group By Time Unit mapping panel using timestamp as \(\text{Property}\) parameter, and “years” as \(\text{TimeUnit}\).
CHAPTER 5. TOOL IMPLEMENTATION

5.2.1.5 Group by Time Unit Static

A Group by Time Unit Static mapping uses the following notation:

\[ \text{GroupByTUS} \left( \text{Property}, \text{TimeUnitStatic} \right) \]

\text{Property} is considered an event log property, and \text{TimeUnitStatic} is known as the time unit static selected for the groups’ construction.

We implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.9, which shows an example of the Group By Time Unit Static mapping panel using timestamp as the Property parameter, and “days (days of the week)” as Time Unit Static.

5.2.2 Filtering Mappings Library

We have considered four filtering mappings: General filter, Numeric filter, Date filter and Date Comparator.
5.2.2.1 General Filter

A General Filter mapping uses the following notation:

\[ \text{General} \left( \text{Property}, \text{TargetValue} \right) \]

\( \text{Property} \) is considered an event log property or a group name, and \( \text{TargetValue} \) is defined as the set of desired values.

We implemented a configuration panel based on the previous notation. This is illustrated in Figure 5.10, which shows an example of the General Filter mapping panel using \text{concept: name} as \( \text{Property} \) parameter, and the set \{“Aanvraag vergunning”, “Vergunning”\} for the \( \text{TargetValue} \) parameter. Additionally, we use a selectable list of values according to the selected \( \text{Property} \).

![Generic Filter (By Selection) configuration panel.](image)

5.2.2.2 Numeric Filter

A Numeric Filter mapping uses the following notation:

\[ \text{NumericF} \left( \text{Property}, \text{TargetValue}, \text{ValueC} \right) \]

\( \text{Property} \) is considered an event log property or a group metric, and \( \text{TargetValue} \) is defined as the desired value, and \( \text{ValueC} \) is known as a value comparator.

Additionally, the configuration panel of this mapping and the relationship with the previous notation is illustrated in Figure 5.11. In this figure we can see an example that filters events using \text{Costs} as \( \text{Property} \) parameter, “>” as \( \text{ValueC} \) and “100” as \( \text{TargetValue} \).
5.2.2.3 Date Filter

A Date Filter mapping uses the following notation:

\[ \text{DateFilter}(\text{Property}, \text{From}, \text{To}) \]

Property is considered an event log property, and From and To are known as Date values used to define the boundaries of the filter.

Additionally, the configuration panel of this mapping and the relationship with the previous notation is illustrated in Figure 5.12. This figure shows an example of the Date Filter mapping panel using the timestamp as a Property parameter, “2011-01-01 00:00:00” for From parameter and “2011-03-01 00:00:00” for To parameter. Additionally, we can find the button “Choose Date”, which calls a calendar for the Date value selection, e.g. “2011-03-01”. (please continue to the next page)
5.2.2.4 Date Comparator

A Date Comparator mapping uses the following notation:

$$\text{DateComparator}(Pr1, Dc, Pr2),$$

$Pr1$ and $Pr2$ are considered event Date log properties, and $Dc$ is defined as a date value comparator.

Additionally, the configuration panel of this mapping and the relationship with the previous notation is illustrated in Figure 5.13. In this figure we can see an example that filters events using timestamp as $Pr1$ parameter, “>” as $Dc$ and $dueDate$ as $Pr2$. 

Figure 5.12: Date Filter (By Date) configuration panel.
5.2.3 Measurement mappings

We have considered four measurement mappings: General Measurement, Time Between, Activity Working Time and Activity Working Time per Group.

Additionally, the construction of these mappings differs from the previous ones because we don't consider the concept of building blocks illustrated in Figure 5.4. In this case we directly select a parameter related to an event property. This was done for the measurement mappings in order to facilitate their use, because we assign a metric to a group constructed with the grouping mappings, which already required the selection of an event property. Finally, we don't allow the use of more than one measurement mapping per continuous axis. This is illustrated in Figure 5.5, which shows the configuration panel of the General Measurement mapping. In the figure we have highlighted the inclusion of event property parameters for the mapping configuration.
5.2.3.1 General Measurement

A General Measurement mapping uses the following notation:

\[ GMS(Property,StatisticalF) \]

Property is considered an event log property, and StatisticalF is defined as the statistical function that we want to use to retrieve a metric.

Additionally, the configuration panel of this mapping and the relationship with the previous notation is illustrated in Figure 5.15, which shows the GMS panel that allows us to select process as Property and Average as StatisticalF.

![Figure 5.15: General Measurement (By Measure) mapping configuration panel](image)

5.2.3.2 Time Between

A Time Between mapping uses the following notation:

\[ TimeBetween(Tg1,Tg2,StatisticalF,GranularityV) \]

Tg1 is considered the name of the first activity (e.g. concept:name property value), Tg2 is the name of the second activity, StatisticalF is the statistical function and GranularityV is the granularity value.

Additionally, in Figure 5.16 we illustrate the configuration panel of this mapping considering the previous definition. In the figure we can see an example that calculates the average time expressed in days between the execution of “Aanvraag vergunning” and “Vergunning”.

![Figure 5.16: Time Between mapping configuration panel](image)
5.2.3.3 Activity Working Time

An Activity Working Time (AWT) mapping uses the following notation:

\[ AWT(Property, From, To, Weekends, StatisticalF, GranularityV) \]

*Property* is considered an event *String* property (e.g. concept:name, org:resource, etc.), *h1* and *h2* define the start and end working hour, *Weekends* identifies if weekends are considered in the calculation. *StatisticalF* is the statistical function and *GranularityV* is the granularity value.

Additionally, in Figure 5.17 we illustrate the configuration panel of this mapping considering the previous notation. In this figure we can see an example that calculates the average AWT expressed in years per activity (concept:name) executed, and a working schedule without weekends that goes from 8:00 to 17:00.

---

![Figure 5.16: Time Between mapping configuration panel](image1)

![Figure 5.17: AWT mapping configuration panel](image2)
5.2.3.4 Activity Working Time per Group

The Activity Working Time per Group (AWTG) is an extension of the AWT that considers the events of a group and not the entire log for the activity duration calculation. Additionally, in the previous chapter we used the following notation for an AWTG mapping:

$$AWTG(Property, From, To, Weekends, StatisticalF, GranularityV)$$

In Figure 5.18 we illustrate the configuration panel of this mapping considering the previous definition. In the figure we can see an example that calculates the average AWTG expressed in years per activity executed. This considering a working schedule without weekends that goes from 8:00 to 17:00.

![AWTG mapping configuration panel](image)

**Figure 5.18:** AWTG mapping configuration panel

5.3 Chart Definition

The GPPA allows us to configure chart visual elements like chart's name, axes labels, orientation (e.g. horizontal or vertical) and appearance (e.g. Bar Chart in 3D). The purpose of this is to obtain flexibility and increase the impressibility of the created chart.

To illustrate this, in Figure 5.19 we can see the configuration panel for the chart selection. (Please continue with the next page)
CHAPTER 5. TOOL IMPLEMENTATION

5.4 Chart Visualization

In this section we observed the visualization of the configured chart, for that reason we have considered the use of an implemented Java chart library instead of a self-developed solution. In the current context it's the best option for covering the GPPA visualization requirements. The Java chart libraries available are strong enough for generating pictures with all the mapping and charts components that have been defined in this implementation.

Additionally, we need an Open Source Java library to generate chart images that can be used for the report generation. The Open Source requirement is related to the implementation environment, which is ProM and it works with a GNU Public License (GPL).

5.4.1 Java chart library selection criteria

For selecting the Java chart library we must consider the requirements mentioned above. Furthermore, we also use concepts like the popularity and support of the library, type of charts supported by it and the visualization elements that they offer, e.g. “chart's name and axes configuration, orientations, appearance, etc.” Finally, we have preselected three of the most popular
options available in the market\(^5\): JFreeChart\(^6\), JCharts\(^7\) and Chart 2D\(^8\).

A comparison of these three solutions is shown in Table 5.1. The table presents the selection criteria and how the tools ranked in a *Low/Medium/High* scale. The library that scored best was JFreeChart, hence we use it for the current GPPA implementation.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
<th>JFreeChart</th>
<th>JCharts</th>
<th>Chart2D</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Popularity/Support</em></td>
<td>The popularity and support is measured using the Google Page Rank [30] with the assumption that a higher index is related to a better support.</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><em>Type of Charts</em></td>
<td>Variety of charts supported, the most the best.</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td><em>Visual Components</em></td>
<td>How the library support the configuration of chart elements like axes labels, chart name, scales, etc.</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td><em>Extra Features</em></td>
<td>How the tool allows the definition and manipulation of existing objects, e.g. “Bar Chart with more than Y axis”</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td><em>Exporting Features</em></td>
<td>Formats available for exporting the chart generated, the most the best.</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>

*Table 5.1:* Java chart library selection criteria.

### 5.4.2 Selection Description

JFreeChart is a Java chart library that allows the creation of different type of charts from a dataset. However, it is based on a GNU Lesser General Public License (LGPL), which allows us to use it in proprietary applications. The charts available and their visualization are the strong features of this library, because we can configure axes, series and labels.

---

\(^5\) See http://www.roseindia.net/chartgraphs/open-source-charting-reporting.shtml
\(^6\) http://www.jfree.org/jfreechart/
\(^7\) http://jcharts.sourceforge.net/
\(^8\) http://chart2d.sourceforge.net/
5.4.2.1 Chart Types
JFreeChart supports charts such as XY charts (e.g. line, spline and scatter.), Pie Chart, Gantt Chart, Bar Chart, Histograms and various specific charts like Wind chart, Polar chart and Bubbles chart.

5.4.2.2 Axes
JFreeChart has the possibility of handling multiple axes on a chart if that chart structure allows it. For example in Figure 5.20, we can see a Bar Chart representation with more than one continuous axis, which increases the expressiveness of the analysis. Providing an opportunity to show performance comparisons, which are required by both case studies. Furthermore, the figure presents a comparison between the frequency of the activities and the working time spent on them.

![Activities frequency versus working time.](image)

**Figure 5.20:** Activities frequency versus working time.

5.4.2.3 Series
JFreeChart provides the option to have multiple series highlighted in a chart, which is useful for cross dimensional analysis (e.g. resources versus activities) like the one shown in Figure 5.21. Furthermore, in the figure we can see the frequency of activities per resource.
CHAPTER 5. TOOL IMPLEMENTATION

5.4.2.4 Visualization

JFreeChart handles different scale axes formats and series colors representations, which can be either automatically generated or configured according to the user's requirements. Furthermore, it has different output types for the visualization captured in a chart, such as Swing components, image files (PNG and JPEG), and vector graphics formats. Additionally, we use the image files feature for the GPPA report generation.

5.5 Report Generation

In order to improve the level of communication of providing the process performance results obtained from the GPPA, we considered the generation of automatic reports in the GPPA implementation. This is also related with the need of having a plug-in able in ProM 6 able to generate reports in a standard portable documentation format. Furthermore, we follow a similar criteria as the one explained for the chart visualization, and we use an implemented Java reporting library. We do this because they are strong enough for the construction and generation of a report. Additionally, this library just as the Java chart library must be Open Source.

Finally, one extra requirement is defined: the library selected must be able to create PDF reports from scratch. We choose PDF\(^9\) because it's a standard format that has printing features and format visualization preservation. Additionally, PDF addresses software compatibility issues, e.g. we can

\(^9\) http://www.adobe.com/pdf/

---

Figure 5.21: Frequency of activities per resource

![Frequency of activities per resource](image)

*Create Case* | *Controleer polls* | *Reis* | *Verranger en vervoer* | *Bevestig sluiting*
see a PDF report in Windows, Linux, Internet Explorer, Firefox etc.

### 5.5.1 Java Reporting Library Selection Criteria

For the selection of the Java report library, we analyze the library's popularity and support. Additionally, we also evaluate the features that a library has for displaying graphical elements, e.g. creation of images, text boxes, tables and more. Finally, we have preselected three of the most popular options available in the market: Jasper Reports, JfreeReport and ITEXT.

A comparison of these three solutions is shown in Table 5.2. This table shows the criteria selection and how the tools ranked in a Low/Medium/High scale. The library that scored best was ITEXT, hence we use it for the current GPPA implementation.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
<th>Jasper R.</th>
<th>JfreeReport</th>
<th>ITEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popularity/Support</td>
<td>The popularity and support is measured using the Google Page Rank [30] with the assumption that a higher index is related to a better support.</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Visual Components</td>
<td>How the library support the configuration of PDF reports with figures, colors, lines, etc.</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Extra Features</td>
<td>How the tool allows the definition and manipulation of existing objects. E.g. “PDFReport with index and Page counter”.</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Importing Features</td>
<td>Formats available for importing images like JPG, PNG, etc.</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 5.2: Report Library Selection Criteria
CHAPTER 5. TOOL IMPLEMENTATION

5.5.2 Selection Description

ITEXT is a Java library used for the creation of PDFs. It is highly recognized and used by important institutions such as NASA, United States Postal Service, U.S. Department of Defense, Google and more. Furthermore, with ITEXT we can create PDFs with images, texts, tables, links, and page numbering. It has PDF merging features to fuse PDF files, which is useful for the generation of reports with an index structure.

Additionally, ITEXT handles two main JAVA objects, the Document and the PdfWriter classes, which allows us to create PDFs from scratch or from any available and accessible data source such as databases, XML files, etc. Furthermore, the construction of a PDF is performed by using high-level objects (e.g. Chunk, Phrase, and Paragraph) and low level functionality (e.g. arcs, circles, rectangles and texts).

5.5.3 GPPA Report

In the previous chapter we defined a structure for the GPPA report. This structure was composed by three sections: cover and summary, Index section and Results.

5.5.3.1 Cover and summary

This section is composed by the title, author, date, and summary of the report. Additionally, in Figure 5.22 and Figure 5.23, we show the Cover and summary configuration panel and its visualization in a report generated with the GPPA.

5.5.3.2 Index Section

It is useful for determining the location of a topic in the report; Figure 5.24 shows an example of the format selected for this. In the figure we can see the title of the chart included in the report and the location of it using a page number.

5.5.3.3 Results

This section covers the performance analysis carried out with the GPPA. For that, we have selected three components for describing a chart: title, description, and chart image. Furthermore, in Figure 5.25 we show the Results configuration panel and its visualization in a report generated with the GPPA. The figure shows an example of a Bar Chart that contains the frequency of activities. Additionally, we can use this configuration panel each time that we want to include a generated chart in the report.
Figure 5.22: GPPA Cover

(please continue to the next page)
CHAPTER 5. TOOL IMPLEMENTATION

Figure 5.23: GPPA Summary

(please continue to the next page)
CHAPTER 5. TOOL IMPLEMENTATION

Figure 5.24: GPPA Index

Figure 5.25: GPPA Results
5.6 Findings

This chapter demonstrated the implementation of the GPPA, which is based on the mappings definitions performed with the meta-model for graphical representation of properties of logs. Furthermore, we used configurations panels for the axes, series, filters and mappings. The chart generation was supported by the Java chart library JFreeChart, which allowed us to define chart components and also to display the performance information in three formats: Bar Chart, Line Chart and Pie Chart.

Additionally, for the report generation we selected ITEXT, which is a Java report library that is useful for the construction and generation of PDF files from scratch. Furthermore, we saw the connection between the structure of the report and the report configuration panels implemented on the GPPA.

In conclusion, this GPPA implementation aims to solve the problem statements defined in Chapter 1. Those problems were related to the development of a new plug-in solution in ProM 6, which must be able to perform a generic process performance analysis and generate a consolidated report with the results. Furthermore, we realized how it is possible to generate this performance report but we still need to evaluate the GPPA with the case studies. This is done for testing if the GPPA is able to answer process performance questions from DossierFlow customers. We perform this evaluation in the next chapter.
Chapter 6

Case Studies Evaluation

This chapter shows the case studies evaluation (please follow Figure 6.1), which is based on the analysis of basic performance questions of two processes from two DossierFlow system customers selected by Océ. We perform this analysis in the context of Process Mining and using the GPPA plug-in. Additionally, in Section 6.1 and Section 6.2 we perform the GPPA evaluation, in Section 6.3 we evaluate the usability of the GPPA and in Section 6.4 we show the findings of this chapter and discuss the results obtained in the GPPA evaluations.

Figure 6.1: Research Goals – Case Studies Evaluation.
6.1 Case Study 1: Sales department of Dutch company

The first case study was introduced in Chapter 3 and it refers to the sales department of a Dutch company, which is in charge of processes related to sales management of products, e.g. credit checks, complains handling, management of clients, etc. Furthermore, this DossierFlow customer has requested an analysis of the process *Order*.

Additionally, we have received a DossierFlow database from this customer, containing information about the *Order* process execution during the time period of May 2010 to March 2011.

6.1.1 Case Study 1 evaluation

We perform the case study evaluation using the performance questions gathered in the preliminary analysis. Additionally, we use the GPPA to answer them.

**Question 1: Which is the percentage of resources' participation?**

To answer this question, first we mapped the resources onto the discrete axis. Furthermore, we chose the *General Measurement* for calculating and mapping the frequency per resource onto the continuous axis. Finally, we used a *Pie Chart* for showing the results in a percentage format.

Figure 6.2 shows the resources' participation, where we can see that Rose was the most active resource from a DossierFlow perspective. Additionally, the following plot was used:

\[ P = (C = \text{Pie Chart}, G = \text{GroupGeneral(org:resource)}, F = \text{noFilter}, V = \text{GMS(org:resource,Frequency)}) \]

![Resource Participation](image)

Figure 6.2: Resources' participation for process *Order*.
**Question 2:** Which is the percentage of activities' participation?

To answer this question, first we mapped the activities onto the discrete axis. Furthermore, we chose the General Measurement for calculating and mapping the frequency per activity onto the continuous axis. Finally, we used a Pie Chart to show the results in a percentage format.

Figure 6.3 shows the activities' participation, where we can observe that “Order kan ingevoerd worden” was the activity with the highest value of participation (50%) from DossierFlow perspective. Additionally, the following plot was used:

\[ P=(C=\text{Pie Chart}, G=(\text{GroupGeneral(concept:name)}), F=\text{noFilter}, V=(\text{GMS(concept:name,Frequency)})) \]

![Activities Participation](image)

**Figure 6.3:** Activities' participation for process Order.

**Question 3:** How many cases were created and closed per month in 2010?

To answer this question, first we used the ByTimeUnit to map months onto the discrete axis. Furthermore, we chose the General Measurement for calculating and mapping the frequency of the activities “dossier gecrëerd” (create case) and “dossier afgesloten” (close case) onto the continuous axis. Additionally, we use a filter only to observe events from 2010 in combination with an extra mapping onto the series axis for highlighting the frequency of each activity. Finally, we use a Line Chart for showing the results.

Figure 6.4 shows the number of cases that were opened and closed per month, in the figure we can detect the critical months for the process Order. For example, in the month of December almost 600
cases were created. But, in the month of February almost 350 cases were closed.

The following plot was used:

\[ P=(C=\text{Line Chart}, G=(\text{GroupByTu}(\text{time:timestamp, months})), \]
\[ F=(\text{General}(\text{concept:name}, \{\text{dossier gecreëerd, dossier afgesloten}\})), \]
\[ V=(\text{GMS}(\text{concept:name, Frequency})) \]

**Figure 6.4:** Number of cases created and closed per month.

**Question 4:** Which are the average, standard deviation and maximum values of the cases’ duration per year?

To answer this question, first we used the *ByTimeUnit* for mapping years onto the discrete axis. Furthermore, we chose the *TimeBetween* for calculating and mapping onto the continuous axis the average, standard deviation and maximum of the time between the execution of the activities “dossier gecrëërd” (create case) and “dossier afgesloten” (close case). Finally, we used a *Bar Chart* to show the results.

Figure 6.5 shows the average, standard deviation and maximum value of the cases' durations. In the figure we can observe that in 2010 the average, STD and Max were 15, 25, and 140 days. But in
2011 these values were 28, 22 and 80 days. Finally, we can conclude that the average duration of a case is not predictable, because there is a high deviation of 13 days between the cases done in 2010 and in 2011.

The following plot was used:

\[ P = (C=\text{Bar Chart}, G=(\text{GroupByTu(time:timestamp, years)}), F=\text{noFilter}, V=(\text{TimeBetween(dossier gecreëerd, dossier afgesloten,Average,days)}, \text{TimeBetween(dossier gecreëerd, dossier afgesloten,STD, days)}, \text{TimeBetween(dossier gecreëerd, dossier afgesloten,Max, days)})) \]

Figure 6.5: Average, STD and Max Case duration

**Question 5:** How predictable is the working time spent in “Behandelen” in 2011 considering the average, standard deviation and maximum values?

To answer this question, first we made an assumption. We considered that an activity is predictable when it has a low standard deviation and a not so high maximum value in comparison with the average value. Additionally, we mapped the activities onto the discrete axis. Then we chose the average working time for calculating and mapping the average, STD and Max of the time spent in “Behandelen”, onto the continuous axes. Furthermore, we used a filter just to observe the events of 2011. Finally, we used three Bar Charts to show the results.

Figure 6.6 shows the average, Figure 6.7 the standard deviation and Figure 6.8 the maximum value
CHAPTER 6. CASE STUDIES EVALUATION

of the activities' working time. In the figures we can see that “Behandelen” has a value that normally goes from 2.4 until 6 days. Additionally, there is a critical case that took 14 days. In conclusion, we can say that in general, to execute “Behandelen” we need a week for simple cases and for complex cases two weeks tops.

The following plot was used for Figure 6.6:

\[
P = (C=\text{Bar Chart}, G=(\text{GroupGeneral}(\text{name})), \\
F = (\text{DateFilter}(\text{time}: \text{timestamp}, '2011-01-01 00:00:00', '2011-04-01 00:00:00')), \\
V = (\text{AWT}((\text{name}, 8, 17, \text{false}, \text{Average}, \text{days})))
\]

![Activities Average AWT](chart.png)

**Figure 6.6**: Average working time of Behandelen.

The following plot was used for Figure 6.7:

\[
P = (C=\text{Bar Chart}, G=(\text{GroupGeneral}(\text{name})), \\
F = (\text{DateFilter}(\text{time}: \text{timestamp}, '2011-01-01 00:00:00', '2011-04-01 00:00:00')), \\
V = (\text{AWT}((\text{name}, 8, 17, \text{false}, \text{STD}, \text{days})))
\]
CHAPTER 6. CASE STUDIES EVALUATION

Figure 6.7: Standard deviation of the working time of Behandelen.

The following plot was used for Figure 6.8:

\[ P=(C=\text{Bar Chart}, G=(\text{GroupGeneral(concept:name)}), \]
\[ F=(\text{DateFilter(time:timestamp,'2011-01-01 00:00:00','2011-04-01 00:00:00'))}, \]
\[ V=(\text{AWT(concept:name,8,17,false,Max,days)}) \]

(please continue to the next page)
Question 6: Do I have cases in 2011 that had an average time higher than seven days between the completions of “Behandelen” and “Order kan ingevoerd worden”? 

To answer this question, first we mapped process instances onto the discrete axis. Furthermore, we chose the TimeBetween for calculating and mapping the time between the executions of the activities “Behandelen” and “Order kan ingevoerd worden”, onto the continuous axes. Additionally, we used a filter for the TimeBetween to only plot the cases that had a value higher than seven days. Additionally, we used a filter to just observe the events of 2011. Finally, we used a Bar Chart to show the results.

Figure 6.9 shows the cases that had an average value higher than seven days. Additionally, this metric is used by the sales department of the Dutch company to validate a predefined standard. In the figure we can see that the standard was not accomplished 11 times, and two of those cases had a value higher than 50 days.

The following plot was used for Figure 6.9:

\[ P=(C=Bar\ Chart,G=(GroupGeneral(concept:instance)), \]
\[ F=(NumericF(TimeBetween(Behandelen, Order\ kan\ ingevoerd\ worden, Average, days),7,>)), \]
\[ V=(TimeBetween(Behandelen, Order\ kan\ ingevoerd\ worden, Average, days)) \]

Figure 6.8: Maximum working time of Behandelen.
Question 7: How many cases in 2010 and 2011 had activities that didn't accomplish the predefined duedate?

To answer this question, first we used the ByTimeUnit for mapping years onto the discrete axis. Furthermore, we chose the General Measurement for calculating and mapping the number of cases per year, onto the continuous axes. Additionally, we used a DateComparator filter to be able to detect the activities executions that had a timestamp higher than their due dates. Finally, we used a Bar Chart to obtain the results.

Figure 6.10 shows the number of cases per year that had activities with delay. In the figure we can see that in 2010 there were 82 cases and that in 2011 the number increased to 94. In conclusion, we can say that the sales department hasn't defined the deadlines correctly or that the activities were constantly executed with delay.

The following plot was used for Figure 6.10:

\[ P = (C = \text{Bar Chart}, G = (\text{GroupByTu(time:timestamp, years})), F = (\text{DateComparator(time:timestamp, >, dueDate})), V = (\text{GMS(concept:instance,SUM)}) \]
Question 8: Which resource had the highest participation as main handler of the cases of question 7?

To answer this question, we only added a series axis that uses a GroupGeneral mapping with the Property parameter mainHandler.

Figure 6.11 shows the number of cases per year that had activities with delay per main handler. In the figure we can see that Kevin had the highest participation in those two years. He had 34 cases in 2010 and 20 in 2011. Additionally, the sales department can use this information to discover the reason of the activities with delay.

The following plot was used for Figure 6.11:

\[
P=(C=\text{Bar Chart}, G=(\text{GroupByTu} (\text{time:timestamp, years}), \text{GroupGeneral} (\text{mainHandler})), F=(\text{DateComparator} (\text{time:timestamp, >, dueDate})), V=(\text{GMS} (\text{concept:instance, SUM})))
\]
CHAPTER 6. CASE STUDIES EVALUATION

6.2 Case Study 2: Dutch Municipality

The second study case was introduced in Chapter 3 and is a Dutch municipality, which is in charge of processes that handle taxes, environmental permit applications, event organizing approval, etc. Furthermore, this DossierFlow customer has requested an analysis of the process “Evenementenvergunningen “(Event Permits).

Additionally, we received the DossierFlow database from this customer, containing information about the Event Permits process execution during the time period of May 2010 to March 2011.

6.2.1 Case Study 2 evaluation

We perform the study case evaluation using the performance questions gathered in the preliminary analysis. Additionally, we use the GPPA to answer them.

**Question 1:** Which is the percentage of resources' participation?

To answer this question, first we mapped the resources onto the discrete axis. Furthermore, we

---

**Figure 6.11:** Number of Cases with delay per year per main handler.
chose the General Measurement for calculating and mapping the frequency per resource, onto the continuous axis. Finally, we used a Pie Chart to demonstrate the results in a percentage format.

Figure 6.12 shows the resources' participation, where we can observe that Peter (50%) was the most active resource from a DossierFlow perspective. Additionally, the following plot was used:

\[ P=(C=\text{Pie Chart}, G=(\text{GroupGeneral}(\text{org:resource})), F=\text{noFilter}, V=(\text{GMS}(\text{org:resource}, \text{Frequency}))) \]

![Figure 6.12: Resources's participation for process “Event Permits”](image)

**Question 2:** Which is the percentage of document types' participation?

To answer this question, first we mapped the activities onto the discrete axis. Furthermore, we chose the General Measurement for calculating and mapping the frequency per activity onto the continuous axis. Finally, we used a Pie Chart to show the results in a percentage format.

Figure 6.13 shows the activity participation, where we can observe that “Vergunning” was the activity with the highest value of participation (42%) from DossierFlow perspective. Additionally, the following plot was used:

\[ P=(C=\text{Pie Chart}, G=(\text{GroupGeneral}(\text{concept:name})), F=\text{noFilter}, V=(\text{GMS}(\text{concept:name}, \text{Frequency}))) \]
Question 3: How many cases entered per month in 2010?

To answer this question, first we used the ByTimeUnit for mapping months onto the discrete axis. Furthermore, we chose the General Measurement for calculating and mapping the frequency of started “Aanvraag vergunning” (requests), onto the continuous axis. Additionally, we used a General filter and a DateFilter only to observe this activity in 2010. Finally, we chose a Line Chart to show the results.

Figure 6.14 shows the number of cases that were opened per month, in this figure we can detect the critical months for the process Event Permits. For example, in the month of February almost 48 cases were created.

The following plot was used:

**Figure 6.13**: Document types participation for process “Event Permits”.

Note: In the following questions we use the term activity, instead of document type.
Question 4: Which are the average, standard deviation and maximum of the cases' duration in 2010? (Note: for the municipality, the duration of the case is based on the time between the completion of “Aanvraag vergunning” and the completion of “Vergunning”).

To answer this question, first we used the GroupByTu for mapping years onto the discrete axis. Furthermore, we chose the TimeBetween for calculating and mapping the time between the completion of “Aanvraag vergunning” and the completion of “Vergunning”, onto the continuous axes. Additionally, we used General filters just to observe the events where those activities finished in 2010. Finally, we used a Bar Chart to show the results.

Figure 6.15 shows the average, STD and Max value of the time between the completion of “Aanvraag vergunning” and the completion of “Vergunning”. In the figure we can see that it's averagely taking almost 8.5 weeks with a standard deviation of 5.5 weeks and a maximum value of 37.5 weeks. Additionally, we conclude that the time spent between these two activities is not predictable, because the difference between the average and standard deviation is too high with a value of 3 weeks.

The following plot was used for Figure 6.15:
Question 5: Which are the average, standard deviation and maximum values of the time between the completion of “Ontvangstbevestiging” and “Verguning”?

To answer this question, first we used the GroupByTu to map years onto the discrete axis. Furthermore, we chose the TimeBetween for calculating and mapping the time between the completion of “Ontvangstbevestiging” and the completion of “Verguning”, onto the continuous axes. Additionally, we used General filters just to observe the events where those activities finished in 2010. Finally, we used a Bar Chart to show the results.

Figure 6.16 shows the average, STD and Max value of the time between the completion of “Ontvangstbevestiging” and the completion of “Verguning”. In this figure we can see that it’s averagely taking almost 9.4 weeks with a standard deviation of 5.3 weeks and a maximum value of 37.6 weeks. Additionally, we conclude that the time spent between these two activities, is not
CHAPTER 6. CASE STUDIES EVALUATION

predictable, and because the difference between the average and standard deviation is too high with a value of 4 weeks.

Finally, the municipality has a standard average value of six weeks for the time between the completion of “Ontvangstbevestiging” and “Vergunning”. In this figure we can observe that was not accomplished.

The following plot was used for Figure 6.16:

\[ P=(C=\text{Bar Chart}, G=(\text{GroupByTu}(\text{time:timestamp, years})), \]
\[ F=(\text{DateFilter}(\text{time:timestamp, }'2010-01-01 00:00:00' , '2010-12-31 23:59:59'), \]
\[ \text{General}(\text{lifecycle:transition, \{complete\}})), \]
\[ V=(\text{TimeBetween}(\text{Ontvangstbevestiging, Vergunning, Average, weeks}), \]
\[ \text{TimeBetween}(\text{Ontvangstbevestiging, Vergunning, STD, weeks}), \]
\[ \text{TimeBetween}(\text{Ontvangstbevestiging, Vergunning, Max, weeks})) \]

**Figure 6.16**: Time Between Ontvangstbevestiging and the completion of Vergunning.

**Question 6**: Considering question 5, which cases had a value higher than the average + the standard deviation?

To answer this question, first we mapped process instances onto the discrete axis. Furthermore, we
chose the \textit{TimeBetween} for calculating and mapping the time between the completion of “Ontvangstbevestiging” and the completion of “Vergunning”, onto the continuous axes. Additionally, we used \textit{General} filter just to observe the events where those activities finished in 2010. Here, we added an extra filter to observe the cases that had an average value higher than 14.7 (9.4+5.3) weeks. Finally, we used a \textit{Bar Chart} to demonstrate the results.

Figure 6.17 shows the cases that had a value higher than \textit{the average + the standard deviation}. Additionally, we can see that 19 cases had a \textit{TimeBetween} value out of the general trend. This metric is useful for the municipality, because they will use those cases to analyze the causes of an average time value higher than 6 weeks.

The following plot was used for Figure 6.17:

\[P=(C=\text{Bar Chart}, G=(\text{GroupGeneral(concept:instance)}), F=(\text{NumericF(TimeBetween(Ontvangstbevestiging, Vergunning, Average, weeks), 14.7, >))), V=(\text{TimeBetween(Ontvangstbevestiging, Vergunning, Average, weeks)})\]

\textbf{Figure 6.17:} Cases with a time over the standard between Ontvangstbevestiging and Vergunning.
6.3 Usability

Once we have analyzed the GPPA for answering process performance questions, we also perform an evaluation of its usability considering the heuristic approach used for the BPA evaluation. This considers criterion such as: Visibility of system status, User control and freedom, Consistency and standards, Error prevention, Recognition rather than recall and Error recognition.

Additionally, we used a Low/ Medium /High scale to evaluate the aspects mentioned above. The GPPA usability evaluation is illustrated in Table 6.1, where we can observe that the values obtained by the GPPA are better than the ones obtained with the BPA but still too low in some criterion, e.g. Consistency and standards and Recognition rather than recall.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility of system status</td>
<td></td>
<td>X</td>
<td></td>
<td>The GPPA gives information about the system status using a progress bar, but this progress bar sometimes remains in 50%, because the progress calculation was made in stages and not in the time elapsed.</td>
</tr>
<tr>
<td>Consistency and standards</td>
<td>X</td>
<td></td>
<td></td>
<td>In general the parameter labels of mapping configuration panels and charts are understandable, but there is not support of the GPPA for automatically suggest labels for the chart's axes. In addition, the configuration of charts, axes and mappings; is not intuitive for new GPPA users.</td>
</tr>
<tr>
<td>User control and freedom</td>
<td></td>
<td>X</td>
<td></td>
<td>At any point of the mappings generation we are able to stop the process. We only found a problem when a chart was already constructed but not yet displayed, because only in that moment we couldn't stop the chart generation.</td>
</tr>
<tr>
<td>Error prevention</td>
<td></td>
<td>X</td>
<td></td>
<td>The system gives alerts about possible errors like incomplete information and not correct parameter configuration. We only detected a problem at the moment to use numerical values because the GPPA didn't control the use of characters instead of numbers, which at the end generated an empty chart.</td>
</tr>
<tr>
<td>Recognition rather than recall</td>
<td>X</td>
<td></td>
<td></td>
<td>Most of the calculations are done each time that a chart is configured. In addition, the GPPA continuously needs to call procedures for retrieving values. This was partially alleviated in the report generation because in that case we keep the charts obtained in a PNG format.</td>
</tr>
</tbody>
</table>
6.3 Findings

In this chapter we evaluated the GPPA with the performance questions gathered in the preliminary analysis. During this evaluation, the GGPA could answer all of these performance questions, which was not possible with the BPA from ProM 5. Additionally, we were able to find interesting information from the processes Order and Event Permits. For example, we were able to observe the cases that didn't accomplish predefined standards; also we were able to detect the cases that were executed with delay in the process Order.

Additionally, we could observe performance information during time periods, which is useful for detecting trends. For example, we could observe the incoming and outgoing cases per month. This information is useful for detecting the critical months, which are related to the peaks of the trend.

Finally, the GPPA is flexible enough to analyze event logs of processes that are executed with a Case Handling orientation like in DossierFlow. But it also showed usability problems that made it complex to use for the first time.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Most of the calculations are done each time that a chart is configured. In addition, the GPPA continuously needs to call procedures for retrieving values. This was partially alleviated in the report generation because in that case we keep the charts obtained in a PNG format.</td>
</tr>
<tr>
<td>X</td>
<td>The GPPA retrieves an error message when a chart was empty or when a chart cannot be generated. In addition, it also retrieves messages in the case of errors in the report generation.</td>
</tr>
</tbody>
</table>

Table 6.1: GPPA usability evaluation
Chapter 7

Conclusions

As discussed in Chapter 1, this master thesis project aims to resolve the following two problem statements:

**Problem Statement 1:** There is not a plug-in available in ProM 6 that allows the execution of a generic process performance analysis with a BI orientation.

**Problem Statement 2:** There is not a plug-in available in ProM 6 that allows the generation of a consolidated report with information obtained through a generic process performance analysis with chart support.

Before we investigated the solution for these problem statements in detail, we observed some preliminary concepts such as BPM, BPMS, BI, Process Mining and ProM. We analyzed that in BPM to be able to redesign operational processes, two trends can be identified: Straight Through Processing (STP) and Case Handling (CH). Additionally, when a CH orientation is taken into consideration, processes are more flexible and variable in their execution. This variability increases the complexity for detecting a process model from an event log, which cannot be used for executing a process performance analysis, e.g. process performance analysis with Petri Nets.

The complexity to detect process models motivated us for taking a simpler but not less effective approach, which aims to retrieve general process performance metrics, e.g. averages, standard deviations of case durations or activities executions. Additionally, the goal of this approach was to *enhance* the process with the support of Business Intelligence (BI). The support of BI is useful for the visualization of the process performance information in an easy to read way (e.g. performance charts), which can be included as well, in a consolidated performance report.

We performed the analysis of this approach in Océ Netherlands DossierFlow. Additionally, we received the event logs of two DossierFlow customers. These customers were the sales department of a Dutch company and a Dutch municipality. Furthermore, we used two processes from these as case studies, the process of *Order* from a Dutch company and the process of *Event Permits* from a Dutch municipality. Although, we saw that in the current implementation of ProM 6, there is not any plug-ins able to perform a generic process performance analysis with a BI orientation. Additionally, we also performed a preliminary analysis of the case studies and gathered process...
CHAPTER 7. CONCLUSIONS

performance questions from the process owners, and we evaluated them with a previous plug-in solution from ProM 5. From the results of this preliminary analysis we confirmed the need of implementing a new plug-in called GPPA (Generic Process Performance Analysis), which must be able to answer the process performance questions in a generic way.

We implemented this plug-in using a meta-model for the graphical representation of the event log properties. This meta-model allowed us to construct a plug-in able to detect process performance metrics with different perspectives and aggregated data visualizations. Furthermore, the GPPA was able to answer the process performance questions, which were helpful to understand how the process is being executed from a DossierFlow perspective. Additionally, we solved the first problem statement, because we contributed with a new option for executing a process performance analysis in ProM.

Finally, the second problem statement was solved by implementing a PDF report that merged the charts captured throughout the use of the plug-in. This was achieved by the support of JFreeChart (Java chart library) and Itext (Java PDF creator library).

In conclusion, the GPPA will contribute to ProM 6 by offering a new highly generic and easy to use analysis approach.

Limitations and Future Work

The current implementation of the GPPA only has three chart types: Bar Chart, Line Chart (Includes Time Series) and Pie Chart. These could cover the analysis of the case studies but the inclusion of other types of chart can be useful for the resource analysis, e.g. Gantt Chart.

Furthermore, we observed some limitations related to the mappings provided, especially with complex metrics such as the Activity Working Time (AWT). This is due to the fact, that with the AWT, we cannot use the same procedure to retrieve the values, and in relation to its calculation method which requires the whole log information. We also detected usability problems related to the user interface and configuration of charts and mappings.

Finally, regarding future work, we suggested the implementation of an XML file to be able to store the configuration done for a particular chart, which must be able to generate automatic reports from previous configurations. We partially implemented this XML feature but we still need to develop an interface for loading and modifying the configurations done for a particular performance chart. Additionally, the user interface of the GPPA can be improved for resolving usability problems and guiding the plug-in users in the configuration of a mapping and a chart.
References


## Appendices

### Appendix A: XES extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Key</th>
<th>Type</th>
<th>Attribute Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>name</td>
<td>string</td>
<td>log, trace, event</td>
<td>Generally understood name.</td>
</tr>
<tr>
<td>Concept</td>
<td>instance</td>
<td>string</td>
<td>event</td>
<td>Identifier of the activity whose execution generated the event.</td>
</tr>
<tr>
<td>Lifecycle</td>
<td>model</td>
<td>string</td>
<td>log</td>
<td>The transactional model used for the lifecycle transition for all events in the log.</td>
</tr>
<tr>
<td>Lifecycle</td>
<td>transition</td>
<td>string</td>
<td>event</td>
<td>The lifecycle transition represented by each event.</td>
</tr>
<tr>
<td>Organizational</td>
<td>resource</td>
<td>string</td>
<td>event</td>
<td>The name, or identifier, of the resource having triggered the event.</td>
</tr>
<tr>
<td>Organizational</td>
<td>role</td>
<td>string</td>
<td>event</td>
<td>The role of the resource having triggered the event, within the organizational structure.</td>
</tr>
<tr>
<td>Organizational</td>
<td>group</td>
<td>string</td>
<td>event</td>
<td>The group within the organizational structure, of which the resource having triggered the event is a member.</td>
</tr>
<tr>
<td>Time</td>
<td>timestamp</td>
<td>date</td>
<td>event</td>
<td>The date and time, at which the event has occurred.</td>
</tr>
<tr>
<td>Semantic</td>
<td>model-Reference</td>
<td>string</td>
<td>log, trace, event, meta</td>
<td>Reference to model concepts in an ontology.</td>
</tr>
</tbody>
</table>

Figure A.1: XES extensions with attributes
Appendix B: Dutch municipality information description

Activity Identification

In order to extract relevant information from the system log of DossierFlow, a different approach was considered. This approach consists in associating the document type of a process, to a stage of it. Furthermore, this approach requires a definition for the time boundaries of the case, because in spite of the registration of the first document and the last document in a specific case, the process can already let us know the duration of it, and it doesn't consider the necessary time for document preparation after creating a case or before closing it. To cover this gap some system activities that are also recorded by Dossierflow will be used such as dossier gecreeerd (case creation), dossier afgesloten (case closed), and dossier hereopened (Case reopened).

Event type definition

Once the activities have been defined, we have to consider which event types will be associated with them. As we could observe, the document types could be considered as process activities, here the identification of the first and last document done for a specific document type will tell us the beginning and ending of a particular activity. Furthermore, this will be the basis for the event type definition, which is shown in Figure A.2, where the first document added with the type A to a case defines the Start of the activity and the last one defines that it's Complete. Additionally, the documents that are not the first or last, are associated with the event type In Process.

![Figure A.2: Event type association](image-url)