An Exploration and Analysis of The Building Permit Application Process in Five Dutch Municipalities

Niels Martin, Gert Janssenswillen, Toon Jouck, Marijke Swennen, Mehrnush Hosseinpour, Farahnaz Masoumigoudarzi

Hasselt University, Agoralaan Gebouw D, 3590 Diepenbeek, Belgium
{niels.martin, gert.janssenswillen, toon.jouck, marijke.swennen, mehrnush.hosseinpour, farahnaz.masoumigoudarzi}@uhasselt.be

Abstract. Similar to other service processes, the process for handling building permit applications is rather unstructured and complex. The goal of this report is to gain insight in the real-life event logs that are provided by five Dutch municipalities as part of the Business Process Intelligence Challenge 2015. Due to the absence of metadata or access to expert knowledge, the PDM methodology is applied to conduct a process analysis. Consequently, three types of analyses are conducted: (i) a control-flow analysis, (ii) a performance analysis and (iii) a role analysis.

Topics: Process Mining, Event Log, Process Diagnostics, Control-flow Analysis, Performance Analysis, Role Analysis

1 Introduction

More and more companies, governmental institutions and other organizations collect large amounts of process-related data in all kinds of information systems. This has led to an explosion of process data. As a result, the process mining field was born as a research domain over a decade ago and produced many techniques which extract process knowledge from process data sets called event logs. This knowledge can relate to the control-flow of the process, its organizational structures, performance characteristics and the case perspective [6;10]. Techniques have been developed to gain insight in the process and, hence, obtain ideas for process improvements which changes the way organizations work [8]. For an extensive overview of the process mining domain, the reader is referred to van der Aalst [8].

This report analyzes the building permit application process in Dutch municipalities. As is the case for other service processes, the considered process is rather unstructured and complex [8]. In this respect, the analysis of event logs can be useful to gain insight in the process. To this end, logs for five distinct Dutch municipalities are explored and analyzed within the context of the Business Process Intelligence Challenge [2]. Given the absence of metadata or access to expert knowledge, the conducted analysis is structured using the PDM methodology [1].

The remainder of this report is structured as follows. The applied research methodology is presented in section 2, followed by an overview of the steps performed to prepare and inspect the event logs in section 3 and section 4, respectively. When the event logs are ready to be analyzed, a control-flow analysis, a performance analysis and a role analysis are performed in section 5, section 6 and section 7, respectively. Finally, section 8 provides the conclusion.

2 Methodological Framework

The results presented in this study are obtained by conducting a process mining analysis, backed by a clear methodology. Three key methodological frameworks have been published to support the execution of a process mining project: (i) the process diagnostic method [1], (ii) the L* life-cycle model [7] and (iii) the PM^2-methodology [7]. The process diagnostic method (PDM) is selected as the project is conducted without access to knowledge of process experts, which is stated to be at the core of the PDM methodology [1]. Conversely, the recently published PM^2-methodology stresses the importance of close cooperation with process experts [7]. The L* life-cycle model encompasses operational support, which is beyond the scope of this report. Given these considerations, PDM is the most consistent with the specifications of the process mining project. It has been successfully applied in its original form or using an extension in several process mining case studies, such as Jans et al. [3] and Rebuge and Ferreira [5].

The original PDM methodology consists of six phases, as visualized in Figure 1. In the first phase, the log needs to be prepared by specifying the definition of a case, identifying activities and verifying whether the meaning of the available timestamps are clear. The latter is complicated due to the absence of metadata or access to knowledge of subject-matter experts.

The second phase involves inspecting the log by generating descriptive statistics such as the number of cases and the mean number of events per case. Using a criterion to distinguish completed cases, incomplete cases can be identified and deleted from the log. This filtered log serves as an input for the remainder of the project.

Next, the PDM framework suggests the execution of a control-flow analysis, establishing the relationships between activities, followed by a performance analysis, e.g. gaining insight in throughput times. In parallel, a role analysis phase can be performed, in which the different roles in the process can be analyzed.

The final phase of the process mining project is the transmission of its results, which will be in the form of this report for this project. Due to the absence of input from a subject-matter expert, Bozkaya et al. [1] state that output interpretation is not part of the project, i.e. the client is responsible for the interpretation of analysis results. However, this work does present some interpretations, making assumptions as a substitute for prior knowledge or metadata.

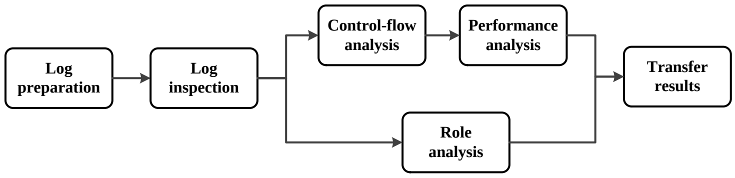

Fig. 1. PDM methodology overview [1]

3 Phase 1: Log Preparation

The log preparation phase involves building of an event log from an information system. As the event logs of the five municipalities are given and no access is provided to the underlying information systems, this phase can be passed.

4 Phase 2: Log Inspection

The log inspection phase firstly generates descriptive statistics to get insight into the event logs provided. Furthermore, the event logs are filtered to prepare them for the analysis phases.

4.1 Getting to Know the Data

Due to the absence of metadata or access to knowledge of subject-matter experts, data exploration was of primary importance to gain insight in the five provided event logs. Data exploration will be helpful during the analysis of specific aspects of the logs later on in the project. The following perspectives were explored in this step:

Activities: How many different activities are present in the logs? What is the overlap between different logs? What are the most frequent starting and ending activities of a case?
Data attributes: What do the data attributes mean? What is the relationship between event labels and codes?
Timestamps: What is the meaning of the different available timestamps? Which timestamps are related to events and which timestamps are related to the case?
Cases: How many case variants (traces) exist within each log? How is the trace length distributed?
Resources: Which different types of resources exist?

The results of exploring all the above mentioned perspectives provided several important insights, which are outlined below. Firstly, the logs of the five municipalities contain events over a period of approximately four years from late 2009 until early 2015, as can be seen in Figure 2.

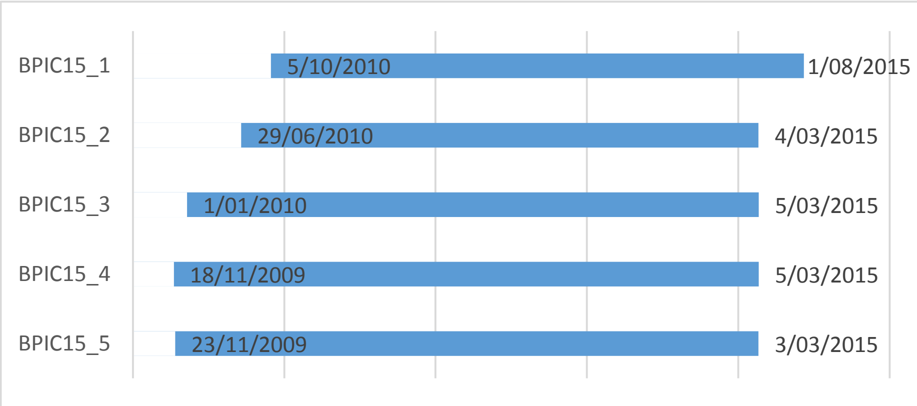

Fig. 2. Timeframe of the five event logs.

Secondly, all logs contain a large number of different activity types (based on the *action_code* attribute), ranging from 356 in municipality 4 to 410 in municipality 2. There exists a large overlap between these activity types over the different logs as shown in Table 1, indicating that similar activities are executed within the municipalities. This seems sensible as the key activities that need to be performed are likely originated from a given legislative framework.

Table 1. Number of activity types shared between the five event logs of the municipalities.

Event log	Shared with Mun. 1	Shared with Mun. 2	Shared with Mun. 3	Shared with Mun. 4	Shared with Mun. 5
Mun. 1	**398**	349	342	326	339
Mun. 2	349	**410**	343	325	331
Mun. 3	342	343	**383**	331	336
Mun. 4	326	325	331	**356**	322
Mun. 5	339	331	336	322	**389**

Thirdly, the original event logs contain 31 data attributes. The absence of metadata and access to expert knowledge renders the use of most of these data attributes complex, as their meaning is unknown. Of key interest are the four data attributes that contain timestamp information, especially *Complete Timestamp* and *dateFinished*. A comparison between these timestamps showed no significant differences between them. Consequently, it is assumed that both timestamps can be used interchangeably. In the remainder of this report, *Complete Timestamp* will be used. Three types of resources are also associated to each event: a resource, a responsible actor and a monitoring resource. Further details on resource presence are presented in Section 7.

Finally, moving from the event level to the case level, the number of traces (or activity sequences) are counted for each log, as shown in Table 2. Almost all cases have unique traces, indicating that a great variety of process behavior is contained in the event log.

From these observations, it can be concluded that the event logs contain behavior of unstructured or spaghetti processes. Therefore, analyzing these logs will be far more challenging compared to analyzing more structured lasagna processes.

Table 2. Number of events, number of cases and number of traces in each event log.

	number of events	number of cases	number of traces
Mun. 1	52217	1199	1099
Mun. 2	44354	832	756
Mun. 3	59681	1409	1202
Mun. 4	47293	1053	912
Mun. 5	59083	1156	1010

4.2 Log Preprocessing

Given the insights from log exploration, the event logs are preprocessed to prepare them for the analysis phase. Essentially this means that the spaghetti process should be made more lasagna-like to enable predictions and recommendations [8].

Key preprocessing efforts are outlined in this subsection. These relate to the creation of new activity identification codes on the one hand and log filtering on the level of subprocesses and cases on the other hand. Note that, unless explicitly stated otherwise, the preprocessed logs have been used in the remainder of this report.

4.2.1 Activity Identification Codes

Firstly, the activities should have a comprehensible meaning. All events in the log can be related to a specific activity that generated it by using the *action code* attribute. This code indicates to which subprocess the activity belongs as well as the phase in which it was executed. However, the codes do not provide insights into what the activities really represent for people not involved in the process. Therefore, the relationship between activity codes and labels was analyzed to see whether the labels can be used instead. Mostly, a one-to-many relationship was found between activity codes and labels. The loss of information by replacing the codes with labels is secondary to the gains in terms of interpretability. Consequently, the new activity identification codes consist of the label of the activity and the name of the subprocess (included in the original activity codes). The addition of the subprocess name connects the labels with the subprocesses to which they are related. For example: '*01_HOOFD_010*' became '*HOOFD register submission date request*'.

4.2.2 Subprocesses

The action codes in the event logs were used to distinguish activities related to the main process (i.e. 'HOOFD') and the different subprocesses. To reduce the complexity of the event log, several subprocesses were replaced by a subprocess start and end event. As such, these subprocesses were collapsed, though not removed from the event log. Based on the first and last activities of these subprocesses at case level, a heuristic was used to learn start and end points of the subprocess and consequently distinguish different instances of the subprocess within the same case. The replacement procedure is independent from other activities that happened in parallel with the subprocess.

Figure 3 illustrates the applied procedure. Assume that the events displayed are describing one single case and a subprocess A is present which consists of three activities A1, A2 and A3. In the first step, all activities of subprocess A are removed. The first and last events are replaced with a start event and an end event of the subprocess. When this would be done for the whole event log, the most common start and end activities of the subprocess at case level can be learned. In step 2, these are used to distinguish different subprocess instances within a case. A new end event was inserted on 7/03/2014, since A2 was considered as an end activity for the subprocess, and it was followed by A1, which was considered to be an start activity. At the latter moment, a new start event was inserted. Note that any intermediate activities which do not belong to the subprocess, have no impact on step 2.

Timestamp	Activity	Subprocess
15/02/2014	H1	
22/02/2014	A1	A
7/03/2014	A2	A
16/03/2014	H2	
17/03/2014	A1	A
4/04/2014	A3	A
5/04/2014	H3	
27/04/2014	A2	A
5/05/2014	H4	

Step 1

Timestamp	Activity	SubProcess
15/02/2014	H1	
22/02/2014	**A Start**	**A**
16/03/2014	H2	
5/04/2014	H3	
27/04/2014	**A End**	**A**
5/05/2014	H4	

Step 2

Timestamp	Activity	Subprocess
15/02/2014	H1	
22/02/2014	A Start	A
7/03/2014	**A End**	**A**
16/03/2014	H2	
17/03/2014	**A Start**	**A**
5/04/2014	H3	
27/04/2014	A End	A
5/05/2014	H4	

Fig. 3. Process of replacing the subprocesses in the event logs.

The replacement of the subprocesses does not imply that their information is completely ignored. Instead, it creates a hierarchy of process models, which allows one to look at the main, high level, process as well as at individual subprocesses. Furthermore, information about the subprocess instance is transferred to the high level log by adding new attributes to the induced start and end events. These attributes contain several characteristics about the specific instance of the subprocess, such as its duration, the activity sequence, whether other activities where interleaving the subprocess instance, etc.

The considered subprocesses are AP, AH I, AH II, OUV, AWB45, BB, BPT, NGV, OPS, VD and VRIJ. Some other subprocesses were not replaced, since they are related to only a few activities or occurred only in a minority of cases. Replacement would therefore not yield any reduction in complexity. The subprocess AWB45 WAW only referred to three events. The subprocesses LGSV, LGSD, CRD, DRZ, EIND and GBH were not considered as they mostly contained only one single activity, with only a few exceptions. Finally, the subprocess OLO was not considered for replacement either, because its activities belonging were found to be widely spread over the cases with a lot of intertwined activities and subprocesses. Therefore, condensing OLO would leave out too much information.

4.2.3 Cases

A final preprocessing step involves filtering the event logs to remove incomplete cases. Including incomplete cases could bias the comparison of the control-flow and throughput times within and among logs.

The case attribute *'case status'* is assumed to distinguish between complete and incomplete cases. The event logs are filtered accordingly, leaving out cases that did not have the value 'G' or 'gesloten'.

5 Phase 3: Control-flow Analysis

After inspecting and preparing the event logs for analysis, phase three of the PDM methodology can be performed in which the control-flow aspect of the process is analyzed. Many algorithms to discover process models from event logs are available, as can be seen in [10]. However, these discovery steps often result in spaghetti-like process models due to exceptions and infrequent behavior captured in the event log. Therefore, the first step in the control-flow analysis phase concerns filtering the event log. Next, the process models for the different municipalities will be discovered from the filtered logs and compared to each other based on their control-flow. After that, other aspects of the control-flow such as the trace length, the number of repetitions and the presence of certain subprocesses are analyzed in order to find differences between the five municipalities. A final subsection concludes the control-flow analysis phase.

5.1 Control-flow Comparison of the Process Highway

5.1.1 Filtering the Logs

Mining a process model for each of the full logs yields spaghetti models that are hard to interpret and compare. In an effort to increase the readability of the obtained process models, the event logs are filtered. To this end, the event filter embedded in the ProM framework is used and the 25% most frequent events are maintained. The characteristics of the filtered logs are summarized in Table 3.

Table 3. Overview of the filtered logs for each municipality.

	Number of cases	Percentage of complete cases	Number of different activities	Percentage of all activities
Mun. 1	567	81	9	5.5
Mun. 2	587	78	10	6.1
Mun. 3	1041	78	9	6.1
Mun. 4	455	79	9	6.1
Mun. 5	863	82	8	5.4

The filtered logs have the following six activities in common:

- HOOFD send confirmation receipt
- HOOFD procedure change
- AWB45 Start
- HOOFD enter senddate decision environmental permit
- HOOFD register submission date request
- HOOFD phase application received

5.1.2 Mining Models

Multiple discovery algorithms have been developed over the last decade (see [10] for an overview). Research on the comparative analysis of multiple process discovery techniques has shown that highly complex event logs can become a major problem for the traditional discovery techniques. Therefore, the heuristics miner [11] (a technique that is robust to noise) and the inductive miner [4] (able to handle infrequent behavior) were applied using the default settings. Taking into account both trace fitness and comprehensibility, the models discovered using the inductive miner ranked higher than the models discovered by the heuristics miner.

The resulting models of the Inductive Miner are block-structured. This means that the models can be divided recursively into parts with a single entry and exit point [4]. From now on these parts will be referred to as 'blocks'. The mined models are mainly characterized by a sequence of blocks containing parallelism, loops and skipping tasks[1] constructs. An overview is provided in Table 4.

Table 4. Overview of the mined models for each municipality.

	# blocks in sequence	Parallelism	Loops	Skips	Trace fitness
Mun. 1	2	1	6	0	0.851
Mun. 2	4	2	5	1	0.770
Mun. 3	4	2	5	2	0.817
Mun. 4	4	1	5	1	0.798
Mun. 5	5	2	6	0	0.880

Municipality 1: The model discovered for this municipality, shown in Figure 4, differs the most from the other municipalities as it consists of a sequence of only two blocks, whereas the other models consist of a sequence of length four or five. The first block within the model is a parallel construct containing seven paths. Of these seven paths, four include a loop construct with the possibility to execute the activities in the loop zero or more times. This means that the activities on these seven paths (if executed), can be done in any order. The next and final block in the sequence is another loop with possibility to execute the activity within the loop zero or more times. From this model one cannot draw specific conclusions related to the order of

[1] This invisible task is the single task on one path in a choice construct. If this path is chosen the other path(s) are 'skipped'.

the activities, but it gives an insight into which activities are performed the most in the log (9 activities):

- HOOFD subcases completeness completed
- HOOFD send confirmation receipt
- HOOFD phase application received
- HOOFD enter senddate acknowledgement
- HOOFD register submission date request
- HOOFD enter senddate procedure confirmation
- HOOFD procedure change
- HOOFD enter senddate decision environmental permit (second block)
- AWB45 start

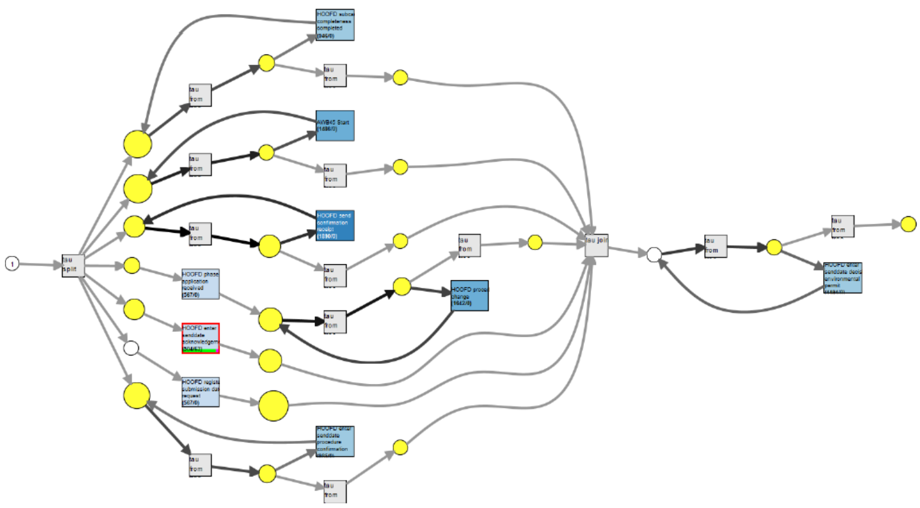

Fig. 4. Process model discovered from the event log of municipality 1.

Municipality 2: The discovered model of municipality 2, depicted in Figure 5, consists of four blocks in sequence. The model starts with a parallel construct containing four paths with four activities:

- HOOFD register submission date request
- HOOFD enter senddate acknowledgement
- HOOFD phase application received
- HOOFD send confirmation receipt

The last activity in the list above is in a loop with the possibility to execute it zero or more times. After the parallel construct in the first block, the second block consists of a choice between a parallel construct and an invisible task in order to skip this part of the model. The parallel construct consist of four paths with four activities:

- HOOFD procedure change
- AWB45 start

- HOOFD creating environmental permit decision
- HOOFD generating decision environmental permit

The first two activities in the list are in separate loops that allow to execute them zero times or more. The third and fourth block of the model each contain a loop of one activity, ‘HOOFD enter senddate environmental permit’ and ‘BB start’ respectively, that offer the possibility to perform them zero or more times.

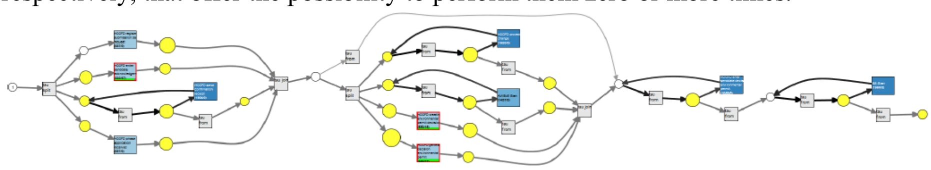

Fig. 5. Process model discovered from the event log of municipality 2.

Municipality 3: The discovered model for municipality 3, visualized in Figure 6, is a sequence of four blocks. The first block contains only one activity: ‘HOOFD register submission date request’ (starting activity). After that, in the second block, two paths within a parallel construct are executed in any order:

- HOOFD phase application received
- HOOFD send confirmation receipt

The last activity in the list above is part of a loop that allows it to be executed zero or more times. Next, the third block offers a choice between a parallel construct and an invisible task in order to skip this part of the model. The parallel construct consists of four paths containing the following five activities:

- AWB45 start
- AH1 start
- HOOFD procedure change
- HOOFD phase advise known
- HOOFD grounds for refusal

The first and the last activity are part of a loop that allows it to be executed zero or more times. The last two activities are situated on the fourth path. Firstly, one can execute ‘HOOFD phase advise known’ and after that one can choose to do ‘HOOFD grounds for refusal’ or skip this activity. Next, the whole path can be repeated. Finally, the last block allows to execute the ‘HOOFD enter senddate decision environmental permit’ activity zero or more times.

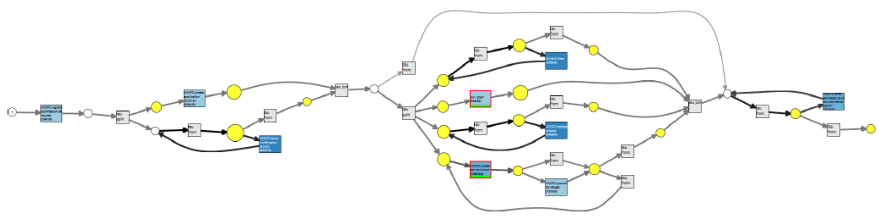

Fig. 6. Process model discovered from the event log of municipality 3.

Municipality 4: The discovered model for municipality four, shown in Figure 7, consists of a sequence of four blocks. The first block comprises a single activity, namely 'HOOFD register submission date request' (starting activity). The next block consists of a parallel construct with four paths and the following six activities:

- HOOFD send procedure confirmation
- AWB45 start
- HOOFD enter senddate procedure confirmation
- HOOFD phase application received
- HOOFD send confirmation receipt
- HOOFD procedure change

Three of the paths contain only one activity (with activity 'AWB45 start' in a loop to be executed one or more times). The other path contains a sequence of 'HOOFD phase application received', followed by a choice between performing 'HOOFD send confirmation receipt' or skipping this and 'HOOFD procedure change'. The first two activities can be executed one or more times (together) and the last activity can be executed zero or more times. The third and fourth block of the model consist each of a loop that allows to execute each of the activities 'HOOFD enter senddate decision environmental permit' and 'BB start' zero or more times.

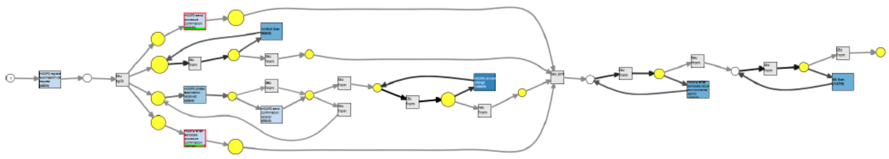

Fig. 7. Process model discovered from the event log of municipality 4.

Municipality 5: The discovered model for municipality five, depicted in Figure 8, is a sequence of five blocks. The first block comprises a single activity, namely 'HOOFD register submission date request' (starting activity). The next block consists of a parallelism construct with two paths each containing the following two activities:

- HOOFD phase application received
- HOOFD send confirmation receipt

The activity 'HOOFD send confirmation receipt' can be executed zero or more times. The third block is again a parallel construct with three paths and the following three activities (in a loop to be executed one or more times):

- AWB45 start
- HOOFD procedure change
- HOOFD subcases completeness completed

Block four and five are equal to model four and consist each of a loop that allows to execute each of the activities 'HOOFD enter senddate decision environmental permit' and 'BB start' zero or more times.

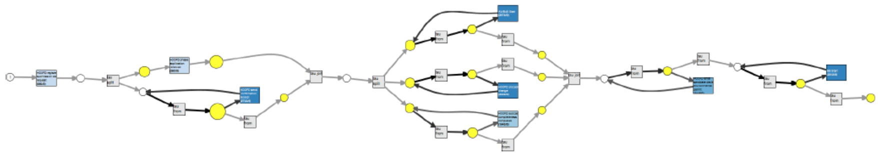

Fig. 8. Process model discovered from the event log of municipality 5.

5.1.3 Comparing the Discovered Models

In this section all five discovered models are compared with each other to see whether there are differences. One difference relates to the activity ‘BB start’, representing the ‘beroep en bezwaar’ (appeal and objection) subprocess. This subprocess appears at the end of the process as shown in the models of municipalities 2, 4 and 5, but does not appear in the models of municipality 1 and 3. This shows that the ‘beroep en bezwaar’ subprocess is not contained within the 25% most frequent events and, hence, is less prevalent in municipalities 1 and 3.

The model of municipality 1 differs the most from the other models as the activities ‘AWB45 start’[2] and ‘HOOFD procedure change’ happen in the first part of the model while they appear in later parts of the models for the other municipalities. Also the model for municipality 1 consists mainly of one large parallel construct (7 paths), while other municipalities contain several blocks in sequence. This could be caused by a less strict order of execution between the most frequent activities in municipality 1 compared to the other municipalities.

Furthermore, the model of municipality 4 differs from the others as it includes a sequence within the parallel paths in the second block of the model. The activities ‘HOOFD phase application received’, ‘HOOFD send confirmation receipt’ and ‘HOOFD procedure change’ mostly happen in this order. However, in the other models, the former two activities are executed concurrently and in the next block of the model the latter activity is performed. This shows that the order of execution between the three activities in municipality 4 is more strict than the order for the same activities in other municipalities.

There are also differences among activities appearing within the parallel paths in each of the models. The models of municipality 2, 3 and 5 each have a few different activities in the second block of the model. This can be explained by the filtering of the logs. If the cutoff percentage (25 % most frequent activities) was slightly increased, the number of the parallel paths within these models increases to include the activities excluded before. For example model two does not include the activities ‘HOOFD phase advice known’ and ‘HOOFD grounds for refusal’, but rather has the activities ‘HOOFD creating environmental permit decision’ and ‘HOOFD generating decision environmental permit’. However, the frequency of the former is only slightly lower than the latter two for municipality 2 (and the other way around for municipality 3). This means that if we increase the cutoff percentage, the two activities previously excluded from model 2 would be included now.

[2] Representing the whole subprocess AWB45, thus making abstraction of individual activities within this subprocess for simplicity.

Remarkably the discovered models of municipality 1 and 2 do not have the activity ‘HOOFD register submission date request’ as the first activity in the sequence, although it is the only start event of the filtered logs. If the discovered model is adapted by putting this activity as the first element of the sequence in each model, the trace fitness remains the same. Yet the interpretation of the model has changed as one can clearly see that each case always starts with ‘HOOFD register submission date request’.

5.2 Trace Length

Moving from the model-level to the case-level, the number of events per case in each municipality can be of interest. Based on the distribution of the number of events per case for each of the five municipalities, which is presented in Figure 9, two groups of cases can be distinguished. A threshold of 40 events was chosen to divide the log of each municipality in two parts, one part with cases containing less than 40 events, and one part with cases containing 40 or more events. These two parts were used to determine whether something has changed in the process that explains this division in short and long cases. In Table 5, some key characteristics for the two logs of the first municipality are given.

Table 5. Log summary for short and long cases in municipality 1.

Log part	Number of events	Number of cases	Number of traces	Average number of activities per case	Average case length (number of events)	Average case duration (in days)
Short cases	4199	210	186	17.542	19.995	75.440
Long cases	25323	486	470	45.901	52.105	105.140

In order to get an insight in the differences between these two groups of cases, an interesting perspective to look at is the frequency of (infrequent) activities in the two groups. For municipality 1, it can be stated that mainly the activities ‘procedure change’, ‘WAW permit aspect’, ‘create publication document’, ‘subcases completeness completed’ and ‘tread subcases completeness’ are much more frequent in the long cases than in the short cases. For example, the activity ‘procedure change’ is executed 131 times in the short cases, or on average 0.62 times per short case, and 1621 times in the long cases, or on average 3.34 times per long case. This means that this activity is executed 5.3 times more in the long cases than in the short cases for this municipality. This indicates that, on average, the procedure has changed more often in longer cases. An overview of all analyses for the other activities can give the domain experts an insight in which activities are causing these differences in case length. Similar calculations for other municipalities generate similar results, but are omitted out of space considerations.

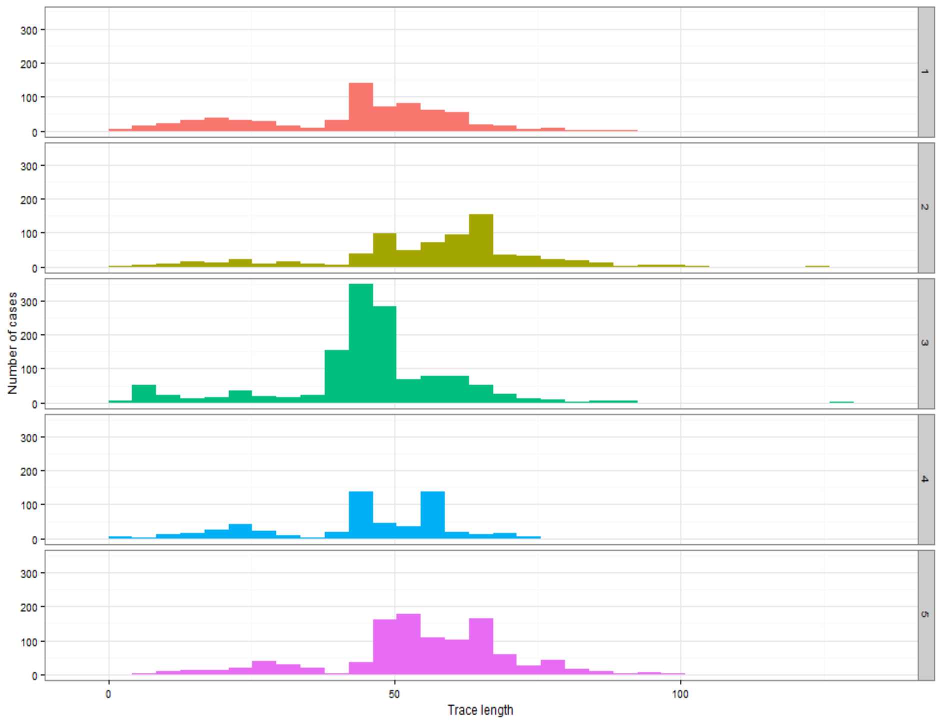

Fig. 9. Distribution of the number of events per case for each of the five municipalities.

5.3 Repetitions

Another aspect that can be of interest for the process of building permit applications is the number of repetitions of an activity. Activities that are executed multiple times after each other or at different locations in a trace reflect inefficiencies in the process and, hence, should be analyzed in detail to identify potential process improvements. From the event log of municipality 1, for example, it can be observed that the maximum number of times that an activity is repeated in one case is7. When looking at the case in which this happens (case 3725165), it can be seen that the activity 'procedure change' is repeated seven times in this case. The same activity is also repeated six times in a case in municipality 2 and 6 times in municipality 4. In Table 6, an overview is provided of the number of occurrences of the most frequent activities in each municipality. For example, activity 'send confirmation receipt' from the HOOFD process is executed on average 1.7 times in a case. This is calculated by dividing the number of occurrences of this activity, 1134, by the number of cases in which this activity occurs, which is 667. From this table, it can be stated that procedure changes occur on average more than once in each case, which can be an indication of inefficiencies in the process.

Table 6. Activity occurrences per case (occurrence/number of cases) in the five municipalities.

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
HOOFD send confirmation receipt	1.700 (1134/667)	1.821 (1320/725)	1.763 (2316/1314)	1.045 (530/507)	1.794 (1792/999)
HOOFD procedure change	1.765 (916/519)	1.808 (1186/656)	1.881 (2094/1113)	1.772 (819/462)	1.801 (1628/904)
HOOFD enter senddate decision environmental permit	1.410 (678/481)	1.602 (1028/642)	1.570 (1647/1049)	1.449 (623/430)	1.503 (1348/897)

Next to this analysis on log-level, the number of repetitions can also be of interest when it is calculated on the activity-level. Therefore, for each municipality an overview is given of the number of occurrences of each activity. The activity 'send confirmation receipt' from the HOOFD process for example, is the activity that occurs the most often in almost all municipalities, except for municipality 4. This activity is repeated around 0.7 times in a case. However, no remarkable differences between the different municipalities can be found from the number of repetitions of the activities in a case.

5.4 Subprocesses

This section discusses the differences in control-flow among municipalities, by focusing on the subprocesses. First, the number of occurrences of the subprocesses are compared. Secondly, the sequences of subprocesses in cases is compared. Finally, the differences of control-flow within the subprocesses are discussed.

Table 7 shows, for each considered subprocess, the percentage of cases they were contained in. For example, subprocess AH I occurred in 71.84 % of the cases in municipality 1.

Table 7. Distribution of subprocesses over the different municipalities.

Subprocess	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
AH I	71.84 %	86.59 %	86.07 %	75.74 %	85.92 %
AH II	43.97 %	61.89 %	56.17 %	46.45 %	57.56 %
AP	1.15 %	1.06 %	6.25 %	0.17 %	1.24 %
AP UOV	0.86 %	0.93 %	0.90 %	0.17 %	0.67 %
AWB45	59.34 %	70.65 %	66.34 %	56.67 %	71.36 %
BB	**18.82 %**	**82.47 %**	**4.14 %**	**66.38 %**	**72.79 %**
BPT	59.77 %	72.91 %	67.32 %	62.74 %	65.37 %
NGV	5.17 %	7.57 %	7.68 %	6.24 %	18.08 %
OPS	13.22 %	11.95 %	13.93 %	23.05 %	25.59 %
UOV	**6.90 %**	**17.26 %**	**7.38 %**	**5.89 %**	**6.76 %**
VD	38.51 %	36.12 %	32.68 %	33.97 %	46.24 %
VRIJ	**15.66 %**	**11.69 %**	**14.46 %**	**19.58 %**	**44.05 %**

Some differences among municipalities can be observed. For instance, the subprocess BB occurred in significantly less cases within municipalities 1 and 3. Subprocess VRIJ occurred remarkably more frequent in municipality 5, while subprocess UOV appeared more in municipality 2.

To understand the high level control-flow between the different subprocesses, Figure 10 shows how the different subprocesses followed upon each other in municipality 1. This figure represents the 8 % most frequent subprocess sequences, which together cover 60 % of the behavior. It can thus be observed that most cases first encounter the subprocess BPT, followed by AWB45. After that, subprocesses AH I and AH II occur most frequently. This sequence of four subprocesses represents 16 % of all cases.

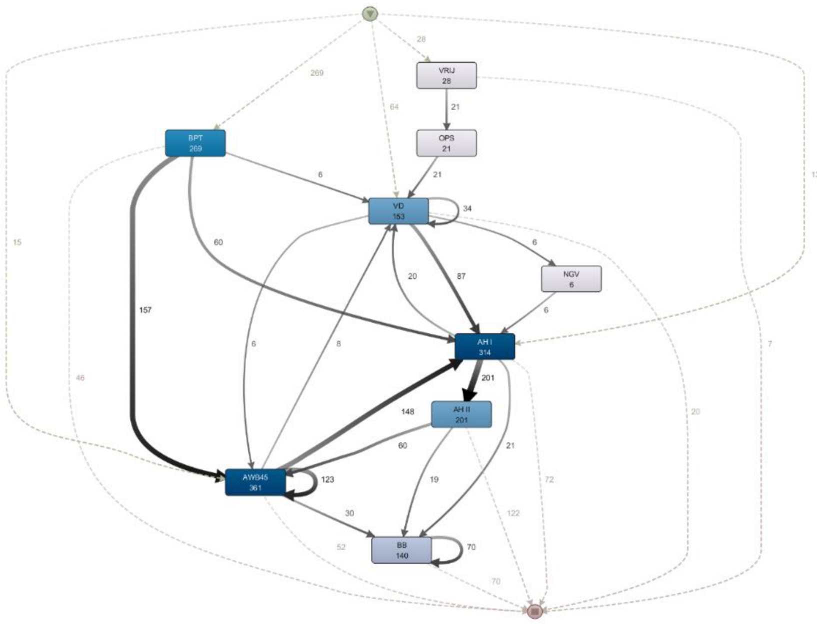

Fig. 10. Sequence of subprocesses in municipality 1.

When comparing the different municipalities, there are ten sequences of subprocesses which are found in all of them. As shown in Table 8, these ten sequences cover about 30 % of the event logs, on average. This means that another 70 % of the behavior in each municipality, on average, is not observed in each of the other municipalities.

Table 8. Coverage of ten common subprocesses sequences.

	Coverage
Mun. 1	30.5 %
Mun. 2	37.2 %
Mun. 3	25.7 %
Mun. 4	38.9 %
Mun. 5	24.4 %

To further delve into this variability, the similarity between pairs of municipalities was investigated. For each pair of municipalities, the common subprocess sequences were identified and their average coverage of the event log was computed. As such, the common sequences among municipalities 1 and 2 covered on average 55 % of the behavior in each of these two logs, as is depicted in Table 9. It can be deducted that, on this high level of abstraction, there are more differences between municipality 3 and 5, while municipalities 1 and 4 share around 60 % of the behavior.

Table 9. Coverage of common subprocess sequences for each pair of municipalities.

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
Mun. 1		55.1%	58.8%	60.4%	37.4%
Mun. 2			45.7%	57.7%	45.0%
Mun. 3				53.9%	28.6%
Mun. 4					47.8%
Mun. 5					

Notice that this analysis only considered the sequences of subprocesses in a case. It did not consider the activities in the main process, nor did it consider the behavior within the subprocesses.

At a lower level, one can look at behavior within each subprocess, and compare it among the different municipalities. For most subprocesses, a common set of activity sequences shared among all municipalities is able to represent more than 90 % of the behavior within each municipality. Notable exceptions are the subprocesses AP, VD and UOV. These are detailed below.

For subprocess AP, Table 7 already showed that significantly more instances are present in municipality 3. Looking at the different activity sequences within this process, there was not one trace which occurred in all the municipalities. This could be attributed to the fact that only one case in municipality 4, and only a few in municipalities 1, 2 and 5 contained an instance of this subprocess. Because of the limited number of observations, similarities are harder to detect.

For subprocess VD, a set of five different activity sequences was found that occurred in each of the municipalities. However, this set only covers around 50 % of the behavior in municipalities 1 and 2. As such, these two contain more behavior, which is absent in the other municipalities. When only focusing on these two municipalities, it is found that they share about 60 % of their behavior, which means that even among these two some variation in control-flow is present.

For subprocess UOV, only five activity sequences were found which occurred in each municipality, and they only managed to cover at most 4 % of the behavior within each of them. Figure 11 shows the control-flow of this subprocess UOV for the different municipalities 1 (left) to 5 (right)[3]. It is clear that the subprocess is the most structured in municipality 1, while more diverse behavior can be observed in municipality 2.

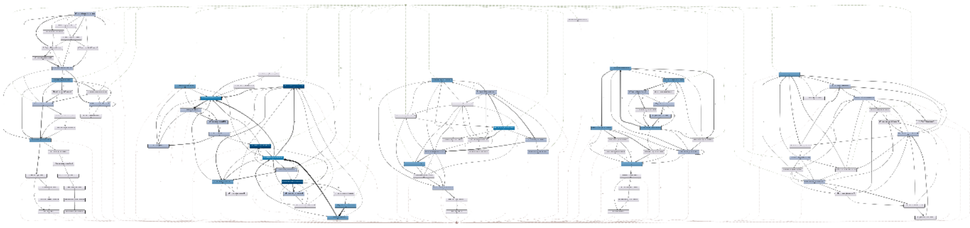

Fig. 11. Control-flow for subprocess UOV in each municipality.

5.5 Discussion

From the control-flow analysis phase of the PDM methodology, it can be concluded that the building permit application process in all five Dutch municipalities under consideration are rather complex and unstructured given the large number of possible sequences of activities. At a rather high level of analysis, some minor differences have been found concerning the structure of the processes. Moreover, a distinction can be made between short and long cases in each municipality, in which the presence of a procedure change in the process plays a role. Next to this, there are differences between the municipalities considering the subprocesses, both in terms of frequency of occurrence and behavior within the subprocesses.

6 Phase 4: Performance Analysis

After discovery of the control flow, the performance of the process can be analysed in the fourth phase of the PDM methodology. In this phase, the event logs are used to give helpful insights in performance questions concerning throughput times in the process. The first subsection presents a general overview of the throughput time of the logs of the five different municipalities. Based on this analysis, two factors that influence the throughput times are investigated in the next subsections. First of all, some of the case attributes in the event logs were found to have an impact on the throughput times. Next to this, the throughput time of the process was also found to be higher in case some of the subprocesses were present. Finally, some general conclusions are drawn based on the dependency of the case attributes and the subprocesses.

[3] The most common activity sequences are shown which together cover 80 % of all the cases.

6.1 General Overview

The throughput times, measured in days, for closed cases was compared for the different municipalities. Their distribution is depicted in Figure 12 and measures of locality and spread are listed in Table 10.

It can be observed that the distributions are very skewed and contain a reasonable amount of outliers. For these reasons, main attention should be directed to the median and the quartiles, since the mean and standard deviation are highly influenced by the outliers and skewness of the distribution.

Table 10. Summary statistics of the throughput time of each municipality.

	Min	Q1	Mean	Median	Q3	Max	St. Dev
Mun. 1	0.357	40.401	96.178	63.000	98.656	1486.000	124.920
Mun. 2	0.000	78.570	159.811	115.000	196.808	1326.000	150.421
Mun. 3	0.003	20.475	62.631	39.000	66.107	1512.000	97.417
Mun. 4	0.000	57.374	110.839	92.635	144.484	882.428	96.477
Mun. 5	1.000	53.373	101.103	79.679	107.586	1344.000	107.190

It is clear that the median duration of cases in municipality 3 is much lower than in the other municipalities. For these cases, 75 % had a throughput time less than 66 days. In municipality 1, 50 % of the cases had a duration of less than 63 days, which is still quite low. Remarkably, for municipality 2, only 25 % of the cases finished within 78 days. It can be concluded that the overall performance based on these results is the best in municipality 3, while municipality 2 scores the worst.

Two factors have been considered to explain the difference in the duration of cases. Firstly, case attributes were looked at with respect to differences in duration. Secondly, also subprocesses were investigated. These can have both an *internal* impact, by taking a lot of time in itself, as well as an *external* impact, by triggering other events which extend or slow down the process. Note that the analyses in the next two sections are based on all the municipalities taken together.

6.2 Case Attributes

Difference in throughput time of cases can be related to case attributes. Of all the attributes, the attributes ‘Includes subCases’, ‘Case Procedure’, ‘Case Parts’ and ‘Request Complete’ show some distinction with respect to durations.

Table 11 demonstrates that throughput times tend to be higher when the case includes sub cases (value ‘J’ for attribute ‘Includes subCases’). This is an intuitive observation as the existence of subcases might point towards a higher complexity of the cases.

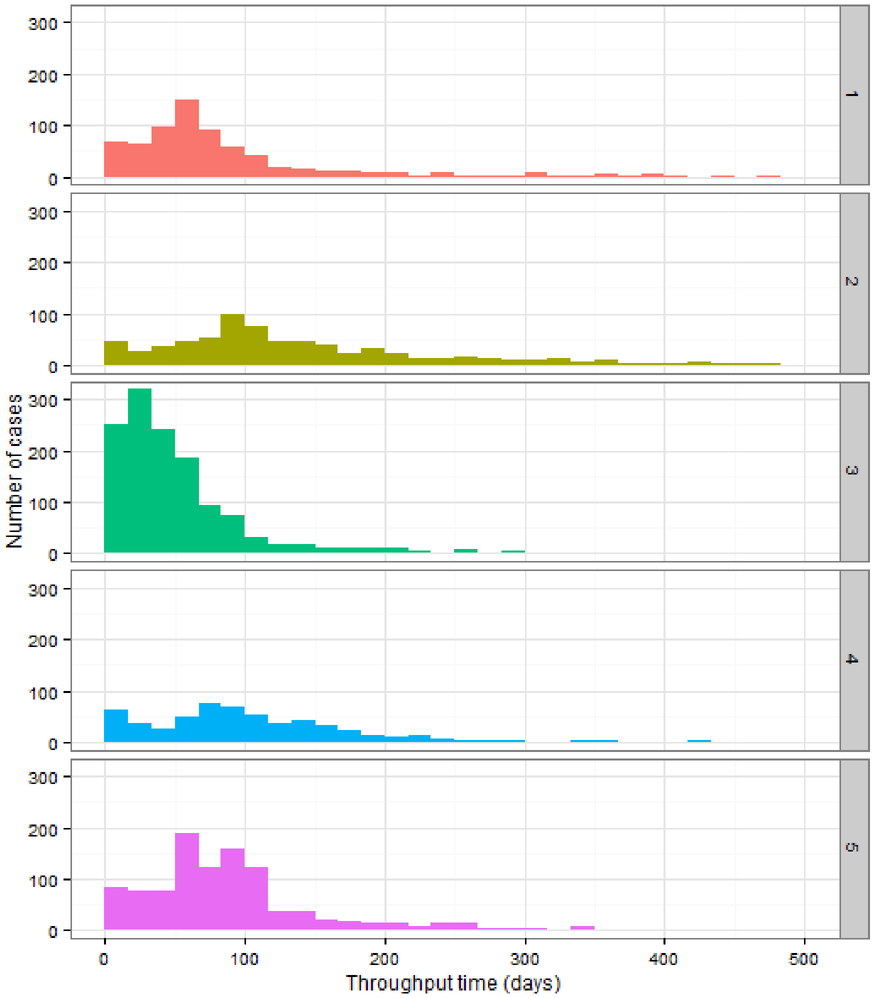

Fig. 12. Distribution of the throughput time of each municipality.

Table 11. Summary statistics for the cases including and excluding subcases.

Includes subCases	Min	Q1	Mean	Median	Q3	Max	St. Dev
J	0.357	55.447	120.257	86.633	134.358	1512.000	132.468
N	0.000	21.510	86.718	53.469	102.000	1260.465	113.103

Furthermore, case that follow the regular (*regulier)* procedure seem to have a lower duration than cases which follow the extended (*uitgebreid)* procedure. However, taking into account the high number of missing values for this attribute, i.e. 88.104 %, this result should be handled with care. This is shown in Table 12.

Table 12. Summary statistics for the cases following the regular or the extended procedure.

Case procedure	Min	Q1	Mean	Median	Q3	Max	St. Dev
Regulier	3.000	50.795	110.696	81.895	114.157	1096	132.045
Uitgebreid	1.159	126.915	256.321	196.475	332.407	1486	205.866

The Case Parts attribute refers to the kind of building permit that is requested. Table 13 lists the performance values for the most common categories, together covering around 80 % of the cases. It can be seen that some permits require more time than others, such as *Milieu (Vergunning)* and *Bouw, Handelen in strijd met regels RO.*

Table 13. Summary statistics for the categories of cases.

Category	Min	Q1	Mean	Median	Q3	Max	St. Dev
Bouw	0.003	38.412	90.583	65.638	103.913	1260.464	104.302
Bouw,Handelen in strijd met regels RO	2.485	55.290	113.791	92.000	126.964	1512.000	135.504
Handelen in strijd met regels RO	1.608	26.000	77.286	58.000	108.000	343.462	61.729
Inrit/Uitweg	4.000	22.000	69.229	36.729	82.346	832.638	106.186
Kap	0.500	17.519	62.002	45.521	87.625	1344.000	74.690
Milieu (vergunning)	0.000	121.442	257.554	217.222	350.750	1486.000	208.716
Sloop	0.000	30.199	69.927	57.472	88.438	365.552	58.001

Finally, the case attribute Request Complete was looked at. This is a binary variable which probably indicates whether the original request was complete or not. Table 14 shows that complete requests tend to result in lower case duration. This seems again logical, as no time is lost while waiting for missing information to complete the request.

Table 14. Summary statistics for the categories of cases.

Request complete	Min	Q1	Mean	Median	Q3	Max	St. Dev
FALSE	2.483	58.717	138.098	92.413	160.000	1260.465	139.507
TRUE	0.000	30.000	91.578	63.759	104.739	1512.000	112.399

6.3 Subprocesses

Of the subprocesses that were explicitly considered during the data preparation, the occurrence of two were found to cause an increase in case duration, namely BB and

UOV. The induced increase in case duration was found to be higher than the average duration of the subprocesses itself.

6.3.1 BB

In Table 15 it can be seen that the cases where the subprocess BB does not occur have a significantly lower duration. Fifty percent of these cases have a throughput time between 21.410 days and 71 days. However, for the cases where it does occur, 50 % of the durations are between 76.675 days and 156.430 days. Remarkably, the average duration of the BB subprocess in a case was only 4 days.

Table 15. Summary statistics for cases with (1) and without (0) subprocess BB.

BB	Min	Q1	Mean	Median	Q3	Max	St. Dev
0	0.000	21.409	67.564	44.458	71.000	1512	100.047
1	0.730	76.675	140.732	101.393	156.430	1486	128.313

6.3.2 UOV

Also the UOV subprocess (see Table 16) seems to have a remarkable influence on the case duration. When it does not appear in a case, the median duration in days is about 63 days. However, when it does, the median duration jumps to almost 223 days. Even though the cases which contain the subprocess are a minority, i.e. only 8.649 % of all cases, the difference in throughput time is statistically significant at the 0.01 level. The increase is remarkably high, since the UOV subprocess was found to have an average duration of only about 50 days.

Table 16. Summary statistics for the cases with (1) and without (0) subprocess UOV.

UOV	Min	Q1	Mean	Median	Q3	Max	St. Dev
0	0.000	32.530	83.173	63.439	100.571	1344	90.194
1	22.000	147.795	278.151	222.731	344.000	1512	208.681

6.4 Discussion

It should be noted that the impact of case attributes and subprocess is not completely orthogonal to each other. In particular, cases that include subcases, have a higher tendency to include subprocess BB. Furthermore, cases that follow the extended procedure are more likely to contain the subprocess UOV. This shows how the characteristics of cases have an impact on the subprocesses *needed* to complete the case, and consequently on the resulting throughput time.

It can be concluded that a reasonable part of the variation in throughput times can be explained by the characteristics attributed to a certain case. When a case has to follow the extended procedure, the subprocess UOV is triggered, which subsequently results in a longer duration of the case. Analogously, when the case includes subcases, it is more likely that the subprocess BB needs to be performed. Since attributes of cases are found to be determinants of throughput times, the large differences between throughput times among different municipalities are unlikely to be the sole result of differences in efficiency, but are also caused by a different composition of the application portfolio the municipality faces.

7 Phase 5: Role Analysis

This section is related to the fifth phase of the PDM methodology, i.e. role analysis. Within the context of the conducted process mining project, this phase involves analyzing the organizational perspective of process mining. This encompasses the analysis of resource-related topics based on an event log [9]. To maximize the available information resources in the log, no filter for complete cases is applied. This filtering is mainly relevant for the analyses related to control-flow and process performance.

To gain a preliminary insight in the resource-related information in the event logs, a general overview is presented in the first subsection. Afterwards, the involvement of resources in the process is analyzed in the second subsection. The third subsection discusses resources that are active in multiple municipalities. The fourth subsection focuses on responsible actors and their responsibility scope. Afterwards, some general observations are summarized.

7.1 General Overview

The event logs contain three different resource types: resource (RES), responsible actor (RA) and monitoring resource (MR). The number of distinct resources per resource type is shown in Table 17. When focusing on type RES, the resources that are assumed to actually execute the activities, municipality 1 possesses 23 distinct RES resources, which is the highest number. However, this does not have to reflect the size of the municipality as staff members might only work part-time. Note that the number of MR exceeds the number of RES for municipalities 1, 3 and 4. The same holds for RA in municipality 3.

To gain insight in the relationship between the resource types, Table 18 investigates their overlap. Panel (a) describes the overlap between resource types. For instance: of the 23 distinct RES resources of the first municipality, 16 of them also occur as a RA resource and 20 as a MR resource. Panel (b) outlines the percentage of events for which the two resource types under consideration are the same. With the first municipality as a notable exception, mainly the large overlap between resource types RA and MR comes forward. Based on this close correspondence, the remainder of this section will focus on resource types RES and RA. Another insight from Table 18(b) is that the responsible actor is actively participating for the cases they are responsible for. Hence, a resource of type RA does not only has a supervisory role, but is also actively involved in the process.

Table 17. Number of distinct resources per resource type

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
Number of distinct RES	23	11	14	10	22
Number of distinct RA	21	7	20	9	8
Number of distinct MR	26	9	22	12	16

Table 18. Resource type relationships.

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
(a) Overlap between resource types					
RES ∩ RA	16	7	10	7	8
RES ∩ MR	20	9	14	8	15
RA ∩ MR	20	7	17	9	8
(b) Percentage correspondence between resource types					
RES ∩ RA	7.426 %	41.571 %	41.858 %	34.825 %	52.368 %
RES ∩ MR	52.285 %	43.873 %	48.973 %	36.168 %	55.315 %
RA ∩ MR	14.667 %	91.465 %	73.707 %	96.475 %	93.044 %

7.2 Involvement of Resources in the Process

To gain insight in the relationship between resource type RES and the process, an analysis is performed highlighting the number of distinct activities a resource executes and the number of subprocesses it is involved in. These results are reported in Table 19, where an activity or subprocess is only recorded when a particular resource executed it at least twice.

The results in panel (a) of Table 19 show that resources tend to execute a great variety of activities. For instance, in the first municipality 169 distinct activities are present and a resource RES, on average, performs 49 of those activities. The percentage number of distinct activities in the event log that a RES resource performs ranges from 28.805 % in the first municipality to 55.135 % in the third municipality. This suggests that the organizational structure is centered around all-around employees, which are supposed to possess all skills and knowledge required to perform a broad range of activities within the process.

Even though RES resources tend to perform a broad range of activities, it should be verified whether these activities are related to a particular subprocess. To this end, panel (b) from Table 19 investigates the number of subprocesses a RES resources is involved in. The results show that RES resources are active within a large number of subprocesses. The percentage of present subprocesses a RES resource is related to ranges from 37.390 % in municipality 1 to 58.714 % in municipality 4. This is consistent with the earlier statement that resources of type RES are skilled and have to be trained to gain insight into the particularities of several subprocesses and their associated activities.

From the previous, it follows that RES resources are, on average, not specialized in a limited set of activities or subprocesses. To verify whether the allocation of RES resources is case based, the mean number of distinct RES resources that are involved in a single case can be determined. Table 20 shows that multiple RES resources are, on average, involved in the process. Note that the presented values are in the same order of magnitude for all five municipalities. A similar conclusion is reached when events to which the RA of a case is associated are not considered. Consequently, case handling is not the unique responsibility of a single RES resource and the RA associated to the case.

Even though the general observations formulated above hold for all municipalities, some differences can be observed. RES resources perform, on average, less distinct activities and are involved in less subprocesses in the first and fifth municipality than in the other ones. Municipalities 1 and 5 are also the largest municipalities in terms of the number of distinct RES resources, as shown in Table 20. Consequently, for larger teams, task division might be somewhat more rigid, leading to resources performing less distinct activities and being involved in less distinct subprocesses.

Table 19. Involvement of RES resources in the process.

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
(a) Activities					
Number in event log	169	164	148	148	148
Number of distinct activities RES performed:					
- Mean	48.680 (28.805 %)	68.500 (41.768 %)	81.600 (55.135 %)	75.250 (50.845 %)	53.840 (36.378 %)
- Minimum	1	4	21	2	1
- Maximum	119	123	112	124	115
(b) Subprocesses					
Number in event log	20	21	20	21	22
Number of distinct subprocesses RES is involved in:					
- Mean	7.478 (37.390 %)	12.270 (58.429 %)	10.380 (51.900 %)	12.330 (58.714 %)	10.150 (46.136 %)
- Minimum	1	3	1	1	1
- Maximum	16	21	17	19	20

Table 20. Number of distinct resources of type RES per case.

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
Mean	2.650	2.490	2.462	2.577	2.809
Minimum	1	1	1	1	1
Maximum	6	6	7	5	6

7.3 Resources Active in Multiple Municipalities

Until now, resources are considered independently for each municipality. However, a limited number of resources execute activities in multiple municipalities. These can be identified in Table 21, which shows that overlapping resources are observed between municipalities (i) 1, 2 and 4, (ii) 2 and 5 and (iii) 4 and 5. The numbers between brackets behind the resource identification code in Table 21 reflect the intensity of the overlap, i.e. the first digit refers to the number of events related to the resource in the 'row' municipality, while the latter considers the 'column' municipality. Note that RES resource with identification code 6 is active within municipalities 1, 3 and 4, but is only associated to a limited number of events. More specifically, this resource only works on a single case within each of these municipalities. Hence, it is disregarded in the remainder of this subsection.

Regarding the activities that the resources related to multiple municipalities perform, an analysis of the involved subprocesses is conducted. Consider for instance resource 560530: it is active in 19 subprocesses in municipality 2 between 29/06/2010 and 03/03/2015 and in seven subprocesses of municipality 5 between 12/05/2014 and 27/02/2015. It can be observed that activity periods in both municipalities overlap, which might indicate that no transfer of resources occurred but collaboration by exchanging resources takes place. Moreover, all subprocesses from municipality 2 are contained in the set of involved subprocesses of municipality 5. Hence, resources might be active in multiple municipalities because of their particular expertise regarding subprocesses. Similar results are obtained for the other resources in Table 21, with resource 560849 as an exception. This resource is involved in subprocess CRD in municipality 5, but was not involved in this subprocess in municipality 4.

Table 21. Resources of type RES active in multiple municipalities (number of associated events in row municipality / column municipality)

	Mun. 1	Mun. 2	Mun. 3	Mun. 4	Mun. 5
Mun. 1	-	none	6 (27/2)	6 (27/4)	none
Mun. 2		-	none	none	560530 (12194/692) 560532 (10104/1337) 560598 (168/1598) 560429 (17/7727)
Mun. 3			-	6 (2/4)	none
Mun. 4				-	560849 (843/156) 560752 (12431/1716)
Mun. 5					-

7.4 Responsibility Scope of Responsible Actors

The prior two subsections focused on resources type RES. To gain insight in resources of the type RA, the relationship between the RA resource type and the case attribute 'parts', which reflects the type of permit that is requested for, is considered.

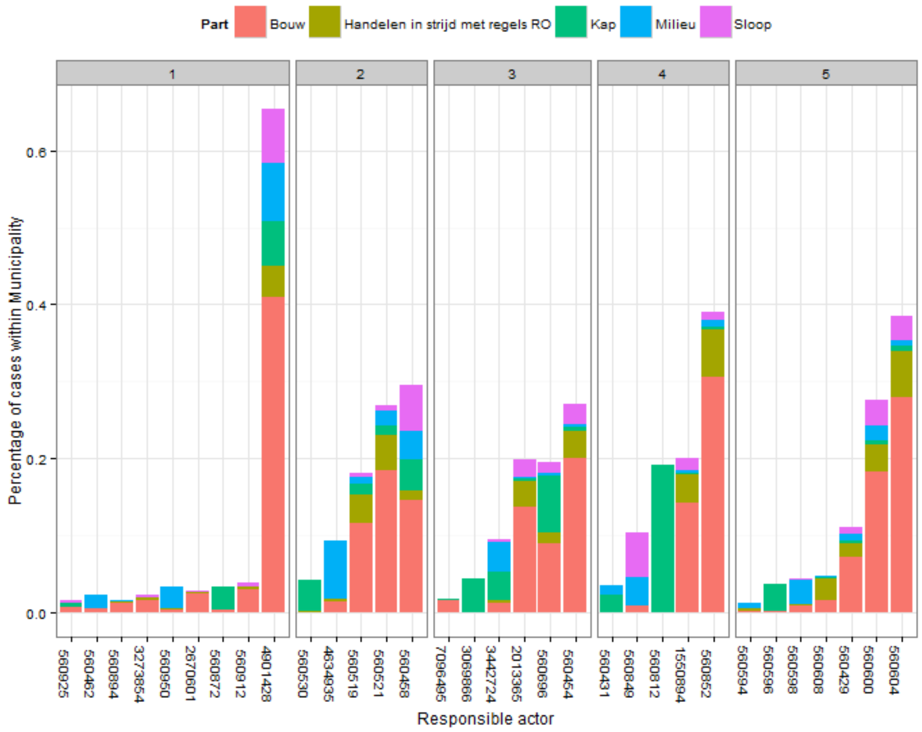

Fig. 13. Responsibility scope RA

Results are reported in Figure 13, which indicates the proportion of all cases in a particular municipality to which a particular RA is associated. Each bar is subdivided in multiple sections based on the attribute 'parts'. To simplify the figure, only the five most frequent values for 'part' are retained, i.e. 'Bouw', 'Milieu', 'Kap', 'Handelen in strijd met regels RO' and 'Sloop'. Moreover, all 'part' attribute values related to 'Milieu' are aggregated, i.e., 'Milieu (melding)', 'Milieu (neutraal wijziging)', 'Milieu (omgevingsvergunning beperkte milieutoets)' and 'Milieu (vergunning)'. To ignore less frequent RAs, an RA is only included in Figure 14 when it is associated to at least one percent of the cases in a particular log.

From Figure 13, it follows that applications within a particular part tend to be spread over multiple RAs. Moreover, RAs are associated to a varying number of cases and cases belonging to several 'parts'. In this respect, 'Kap' is an exception as RAs are present within each municipality whose main responsibility is 'Kap'.

With regards to responsibility sharing, municipality 1 differs from the other municipalities. For instance, the same RA is associated to almost 70 % of all cases, while the other RAs largely focus on a particular type of applications. In other municipalities, applications of a particular 'part' tend to be shared among several RAs.

7.5 Discussion

Based on the conducted resource-related analyses, it can be concluded that resources of type RES do not tend to focus on a limited number of activities in particular subprocesses. In contrast, they often execute a wide range of distinct activities spanning an important part of the municipality's subprocesses. This implies that employees have a broad skill basis as all competencies and knowledge required to perform a broad range of activities within several subprocesses is required. Even though this makes the allocation of resources within a municipality rather flexible, it does not leverage the potential efficiency gains associated to specialization. As the latter might lead to more efficient operations and, hence, lower throughput times, it is worthwhile to investigate the effects on process performance of increased specialization. This can be especially helpful given to the complex and dynamic of the legislative framework which has to be taken into account when judging building permit requests, e.g. regarding tightened environmental requirements. In municipalities 1 and 5, resources are involved in a more restricted set of activities and subprocesses. However, the percentage of distinct activities and subprocesses RES resources are involved in are too high to state that the organizational structure is based on profound specialization.

Besides increased specialization, a more intense collaboration between municipalities is recommended. To date, only a limited number of resources of type RES are active in multiple municipalities. When resources are assigned more flexible across municipalities based on e.g. the cases at hand, this also provides opportunities for specialization as particular activities and subprocesses are executed more often when multiple municipalities are considered simultaneously. This opens potential for efficiency gains, which have to be weighed against the support among staff members to cooperate with other municipalities.

The analysis of the relationship between the RA resource type and the case attribute 'parts' suggests that there is a somewhat clearer responsibility structure for resource type RA than for type RES. E.g.: in all municipalities there is a particular RA whose main responsibility is related to permits of type 'Kap'. This analysis showed deviating behavior for municipality 1 as a single RA is associated to almost 70 % of all cases. For the other municipalities, cases are more evenly spread among a limited set of RA resources.

8 Conclusion

This report analyzed event logs presenting the building permit application process of five Dutch municipalities using the PDM methodology. Firstly, the event logs were inspected by getting to know the data and preprocessing the event logs for analysis.

Secondly, the control-flow of the process was investigated by discovering the process models for each municipality. On a high level of abstraction, only minor differences in the process structures were found. Moreover, a distinction was made between short and log cases and differences between municipalities were found regarding subprocess frequency and behavior.

Thirdly, the performance analysis led to the conclusion that certain case attributes and subprocesses influence the throughput time and, consequently, process performance.

Finally, a role analysis was performed which showed that resources tend to be active in a wide range of activities and subprocesses. Moreover, a limited set of resources were active in multiple municipalities.

This reports provides process experts with factual insights in the building permit application process. To convert these insights into specific recommendations for the process owner, metadata and access to expert knowledge is required.

9 References

1. Bozkaya, M., Gabriels, J., van der Werf, J.M.: Process diagnostics: a method based on process mining. Proceedings of the International Conference on Information, Process, and Knowledge Management, 22--27 (2009)
2. IEEE Task Force on Process Mining. "Business Process Intelligence Challenge (BPIC)." 11th International Workshop on Business Process Intelligence 2015, http://www.win.tue.nl/bpi/2015/challenge
3. Jans, M., van der Werf, J.M., Lybaert, N., Vanhoof, K.: A business process mining application for internal transaction fraud mitigation. Exp Syst Appl 38, 13351--13359 (2011)
4. Leemans, S., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: 34th International Conference Application and Theory of Petri Nets and Concurrency, pp. 311--329. Springer, Berlin (2013)
5. Rebuge, A., Ferreira, D.R.: Business process analysis in healthcare environments: a methodology based on process mining. Inform Syst 37, 99--116 (2012)
6. Rozinat, A., de Medeiros, A.K.A., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The Need for a Process Mining Evaluation Framework in Research and Practice. LNCS 4928, 84--89. Springer, Heidelberg (2008)
7. van Eck, M., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM^2: a process mining project methodology. LNCS 9097, 297--313 (2015)
8. van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer-Verlag, Heidelberg (2011)
9. van der Aalst W.M.P., Adriansyah A., de Medeiros A.K.A., Arcieri F., et al.: Process mining manifesto. LNBIP 99, 169--194 (2012)
10. vanden Broucke, S.: Advances in Process Mining: Artificial Negative Events and Other Techniques. Ph.D. thesis, KU Leuven (2014)
11. Weijters, A., van der Aalst, W.M.P., De Medeiros, A.K.A.: Process mining with the heuristics miner-algorithm, Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1--34 (2006)