BPIC 2015: Diagnostics of Building Permit Application Process in Dutch Municipalities

Irene Teinemaa, Anna Leontjeva, and Karl-Oskar Masing

University of Tartu, Liivi 2, 50409 Tartu, Estonia,
{irheta,anna.leontjeva,kom}@ut.ee

Abstract. Business process mining has been successfully utilized to optimize processes in various organizations. Governmental institutions are no exception — administrative processes are in need for various improvements. In this work, we analyze building permit process in Dutch municipalities from different perspectives. We identify two main roles performed by the employees and propose several improvements on the organizational structure. Additionally, we identify changepoints in the process based on the historical data, as well as propose a simulation framework to assess the effects of possible future changes. Moreover, we characterize differences in terms of throughput times and control flow between the municipalities.

Key words: process mining, role identification, organizational structure, process drift, simulation, performance analysis, control flow

1 Introduction

The Netherlands switched to "All-in-one Permit for Physical Aspects" in 1 October 2010. The goal was to unite, and by doing so, simplify the process of issuing permits related to construction, demolition, renovation, environment and alike. However, the change has not been flawless. There have been complaints regarding long waiting times, which often exceed the official 8 weeks, as set by the General Environmental Law (WABO — Wet algemene bepalingen omgevingsrecht).

This paper analyses the permit issuing process in 5 municipalities, in order to discover the underlying reasons. We are interested in coming up with suggestions to improve the process and, therefore, the throughput times. In order to achieve that, we will try to answer 6 questions posed in the BPI Challenge 2015 which are the following:

1. What are the roles of the people involved in the various stages of the process and how do these roles differ across municipalities?
2. What are the possible points for improvement on the organizational structure for each of the municipalities?
3. The employees of two of the five municipalities have physically moved into the same location recently. Did this lead to a change in the processes and if so, what is different?

4. Some of the procedures will be outsourced from 2018, i.e. they will be removed from the process and the applicant needs to have these activities performed by an external party before submitting the application. What will be the effect of this on the organizational structures in the five municipalities?
5. Where are differences in throughput times between the municipalities and how can these be explained?
6. What are the differences in control flow between the municipalities?

The outcome of different processes is largely determined by the people doing them. For that reason we have to first understand the roles they perform in each municipality. This knowledge allows municipalities to better understand the workflow and, therefore, to organize it in an optimal way.

It is often the case that processes take longer due to some participants having huge workload. In order to solve it, we need to identify the individuals and decide which steps should be taken to improve the organizational structure.

One possible change in municipality's organizational structure would be to share headquarters with another, which is exactly what two of the municipalities under scrutiny have done. In order to see, whether it has had any real effect, we try to identify from the data, which municipalities have moved into the same location and how has it altered their processes.

Another way to reduce the processing time is to outsource some of the procedures. This would require permit applicants to perform the activities by an external party before submitting an application. We are interested in which activities should or could be outsourced and how will it affect the organizational structure.

Although the permit application process in all five municipalities should be identical, it is not the case in practice. With the purpose of understanding where those differences lie, we perform a comparative analysis of throughput times and control flows between municipalities.

Throughout the work, we use various tools that support our analysis, which are briefly summarized in Table 1.

Table 1: Tools used for the analysis

Tool	Purpose
Java	preprocessing the log files
R	various statistical analysis of the data
Disco	analyzing the process models and organizational structures
ProDrift	finding change points in the process
BIMP	simulation
Signavio	modifying the underlying process model for simulation
ProM 5.2	conformance checking across municipalities
ProDelta	comparing the event logs of two municipalities

The rest of the paper is organized as follows. Section 2 gives an overview of the data at hand and discusses our approach to handling several issues with it.

In Sections 3-8 we discuss and analyze each of the 6 topics described above. The findings are concluded in Section 9.

2 Data description and preprocessing

In this work, we analyze the data from 5 anonymous Dutch municipalities from years 2010 to 2015. For brevity, we will refer to the five municipalities as M1 to M5 hereinafter.

The number of cases in each municipality differs from 832 in M2 to 1409 in M3 (Table 2), while the number of events varies from 44354 to 59681.

Table 2: General statistics of the data

	M1	M2	M3	M4	M5
# cases	1199	832	1409	1053	1156
# events	52217	44354	59681	47293	59083

In total, the data sets contain 356 activity names, but have 495 *action codes*. Therefore, some activity names correspond to more than one action code. In general, the action codes consist of three parts: two numbers, followed by a number of characters, followed by three digits. An example activity code is 01_HOOFD_010. According to the description provided with the data, the first two parts of the action code (01_HOOFD_xxx) should indicate the (sub)process the activity belongs to. The last part (xx_xxx_010) should hint on the order the activities are performed in, while the first of these three numbers (xx_xxx_0xx) is supposed to hint on the phase within the process.

In the following paragraphs, we will discuss some of the issues with the data and how we decided to deal with them.

Pattern abstraction. It is evident that in order to comprehend the process, the 356 activities should be somehow abstracted to a higher level. One option would be to take advantage of the action codes. For example, we could state that all activities that start with 01_HOOFD_0xx belong to the same phase, as suggested in the data description. However, this would still leave us with the problem of finding meaningful names for the phases.

In total, there exist 35 different activity names that start with the word "phase", for example *phase application received* or *phase decision taken*. These "phase-activities" seem to wrap up major parts in the process. However, the phase-activities do not correspond directly to the action code phases. For the sake of simplicity, we chose to create a high-level representation of the process by filtering only the phase-activities.

On Figure 1 we see the phase-level process model based on M1, built on very frequent behavior only. The process starts with reception of the application. Additional information is requested if needed. The process proceeds with declaring the application receptive. Then, advice is asked, probably from the stakeholders for the given case. After that the decision is taken, sometimes preceded by the declaration that decision is ready. Finally, the decision is sent.

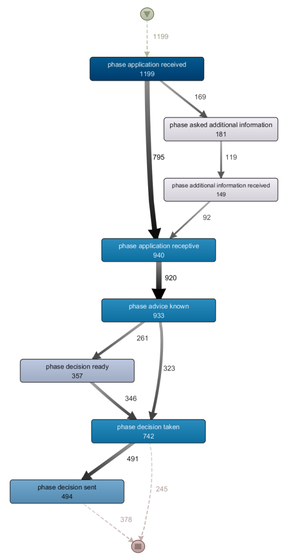


Fig. 1: Phase-level process model for most frequent behavior

In the following, we will switch between two levels of abstraction of the process and refer to them as *low-level activities* (i.e. all of the 356 activity names) and *phase-level activities* (the 35 activities that start with "phase") respectively.

Order of events within a trace. Events in cases tend to happen in "batches" — several events are performed on the exact same timestamp. This means that we cannot rely on the given order of the events, as a lot of events happen in parallel. While some of the methods we use are able to capture the concurrency relationship between two activities, others are not. Thus, we decided to order the events within a trace with the help of the information captured in action codes. More precisely, we order the events within a batch based on the subprocess (e.g. 01_HOOFD_xxx) and then order the events within each subprocess according to the last three numbers (xx_xxx_010).

Completeness of cases. Several methods we use assume that the cases have finished. The ongoing cases may disturb process discovery and analysis of case variants. However, in the given data there seems to be no implicit indication of whether the trace is completed or not. We looked into three case variables that we thought could be related to the completeness of the case: *requestComplete*, *caseStatus*, and the presence of *endDate*. However, by looking at the data, we are not convinced that none of these attributes indicate the completeness of a case. Also, neither the last event nor phase conclusively states whether the case is finished, because the case can end with various different activities. Unfortunately, the data description did not provide any information about case completeness either. Therefore, we decided not to filter the data with respect to case completeness, in order not to lose any valuable data.

Timestamp issues. The data contains several timestamp variables: *time:timestamp*, *dateFinished*, *planned*, *dateStop*, and *dueDate* at the activity-level and *startDate*, *endDate*, and *endDatePlanned* at the case-level. However, most of these attributes turned out to be unreliable. Out of the 8 listed timestamp variables, 5 are missing in a vast number of cases (dueDate, dateStop, planned, endDatePlanned, and endDate). Furthermore, the timestamp of the last event in a case does not usually coincide with the endDate of the case.

At first, we computed the duration for events by taking the difference of dateFinished and time:timestamp. However, the resulting durations were negative in a relatively large number of events, implying that the event was finished before it was started.

As we did not have the possibility to consult with the process owner how the time attributes are reported, we decided to proceed with only one timestamp variable (time:timestamp) for the events. As a result, we compute durations between *event pairs* — each duration comprises the actual time for performing the event and the waiting time until the start of the next event. The event pairs are constructed using any two sequential events, where the events in cases are ordered based on the action codes as discussed above. Event pairs that occur in the same batch are not discarded, but have zero duration.

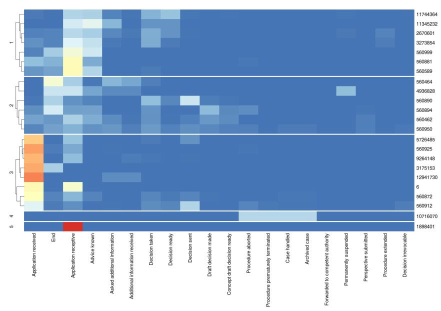
3 Roles and stages

In order to investigate which activities every resource performs, we decided to construct the *activity profiles* for each resource. Namely, for each resource we compute the frequencies of all performed activities.

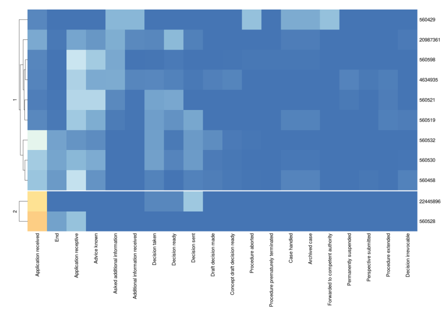
However, the first research question implies that we should identify the *stages* of the process, instead of presenting all the 356 activities. For the sake of simplicity, we decided to use the phase-level activities (see Section 2) interchangeably with the notion of stages. However, simply filtering the phase-level activities would not be a valid approach here, because we would omit the resources that perform only low-level activities. Thus, we mapped each activity to a phase using the following rules.

1. If the action code of an activity has the same phase indicator (e.g. *01_HOOFD_0xx*) as a phase-activity occurring in the same trace, then the activity is directly mapped to this phase-activity. If there are multiple phase-activities with the same phase indicator, then the likelihood of the activity to belong to any of these phase-activities is considered equal.

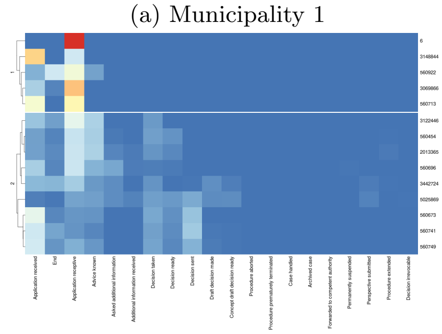
2. If an activity does not have a matching phase indicator, we consider the phases that occur at the same timestamp (same *batch*). The likelihood of the activity to belong to any of the phases occurring in the same batch is considered equal.
3. If there are no phase-activities that occur in the same batch as the given activity, we process the activity together with the following batch(es).
4. Activities that occur at the end of the case, after the last batch containing a phase-activity, are assigned to phase "End".



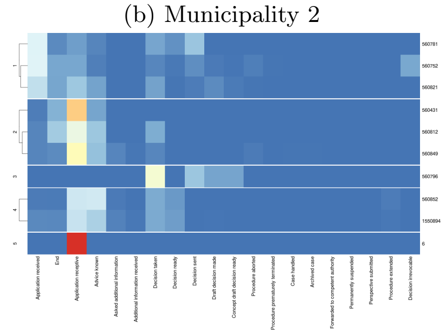
(a) Municipality 1



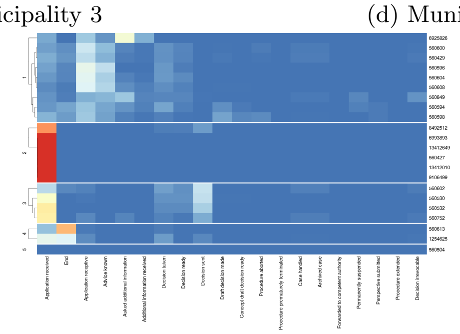
(b) Municipality 2



(c) Municipality 3



(d) Municipality 4



(e) Municipality 5

Fig. 2: Resource-phase heatmaps

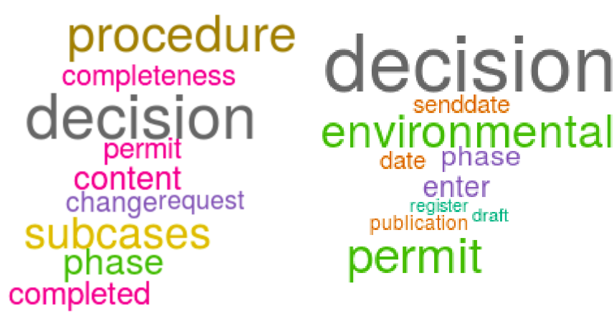
The resulting resource-activity matrices can be seen on Figures 2a-2e, where red color indicates high frequency for a resource to perform a given phase, yellow/light blue indicate medium frequency and darker blue indicates low frequency. The values are scaled by resource, which intuitively highlights the phases in which this resource is more often involved. On the figures, only the top 20 of most frequent phases are shown.

In order to find the *roles* in each municipality, we clustered the resources based on their activity profiles. For this purpose, we used the model-based clustering approach [1]. The optimal number of clusters was chosen using Bayesian Information Criteria and the number ranges from 1 to 5 clusters. The resulting clusters are separated by fine white lines on Figure 2.

In all of the municipalities we discover clusters that are mainly focused on the *Application received* phase. These employees are the first resources to process an incoming case. We can call this role the *administrators*. On the figure, this group is presented as cluster 3 in M1, cluster 2 in M2, cluster 1 in M3, cluster 1 in M4, and cluster 2 in M5. Still, note that the Application received phase can be performed by other resources as well with a lower frequency.

Another pattern that emerges is performance of *Advice known* and *Decision made*, along with other decision-related phases. This indicates that some resources are focused on the actual decision making process more than others. This role could be named *decision-makers*. The clusters who stand out according to these activities are 1 and 2 in M1; 1 in M2; 2 in M3; 1, 2, and 4 in M4. In M5 there seems to be different groups of decision-makers according to whether they also perform Advice known or not (clusters 1 and 3-4 accordingly). The *Application receptive* phase is performed by both roles with high frequency with the exception of M5 where it is mostly performed by decision-makers.

In case of M1 we see that there are two clusters of decision-makers. In order to dive deeper into their differences, we take a look at the low-level activities that are performed by the members of these clusters. On Figure 3 we see the word clouds built for all low-level activities performed by the members of clusters 1 and 2. We see that the second cluster is more focused on environmental permits.



(a) Cluster 1

(b) Cluster 2

Fig. 3: Word clouds of low-level activities in decision-making clusters

M4 stands out by the resources having the administrator role also performing the decision-making role quite frequently. By taking a closer look, we see that also in M1, M2 and M5 at least some administrators can make decisions as well, but without performing Advice known. We can conclude that there are some simpler cases which can be resolved using a simplified procedure (without collecting advice) and the administrators are competent to settle those cases without delegating to decision-makers.

Out of the 5 municipalities, M3 and M2 seem to be very flexible, where most resources can perform most activities, and the separation into roles is less clear. On the contrary, the duties are most clearly separated in M1.

A few one-resource clusters exist in M1, M4, and M5. One reason for those clusters is that some resources are *occasional workers*, who have performed very few activities in total, and their activity profiles are, therefore, unstable. In M1, most of these cases happened near the beginning of the data period in 2011. A possible explanation is that an internal crash course was held at some point in time for using the information system. All employees could have simply tried out the new system, even when they do not use it in their everyday work. Another explanation for occasional workers could be that in some cases a fast response was required and resources who under normal conditions do not use the system, needed to take action.

Seven resources are *cross-municipal resources*, who are involved in more than one municipality. The frequencies of events that are performed by those resources are brought in Table 3.

We can note that in almost all the cases, M5 is one of the municipalities involved in the cross-municipal organization. Thus, M5 could be a higher-level municipal unit, with whom M2 and M4 occasionally need to collaborate with. The two key resources from M2 (560530 and 560532) executed tasks in M5 only in the last two years, 2014 and 2015. This might indicate that the two employees have moved to the same location as M5. From Figure 4 we see that these two resources started working for M5 at the same time — in May 2014. Resource 560532 has continued working for M2 as well, but 560530 has gone over to M5.

There is one exceptional resource (6), who performs few tasks in M1, M3 and M4. This could be a higher level manager.

Table 3: Frequency of events performed by cross-municipal resources

Resource	M1	M2	M3	M4	M5
6	26	0	2	3	0
560429	0	19	0	0	7590
560530	0	11479	0	0	683
560532	0	10080	0	0	1317
560598	0	183	0	0	1737
560752	0	0	0	11948	1676
560849	0	0	0	764	154

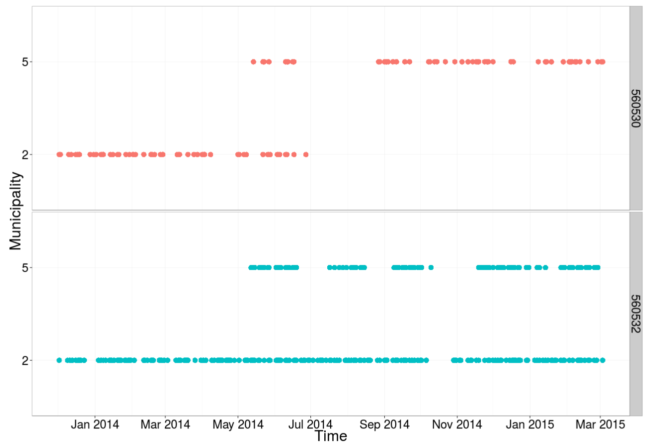


Fig. 4: Time of events executed by the cross-municipal resources in M2 and M5

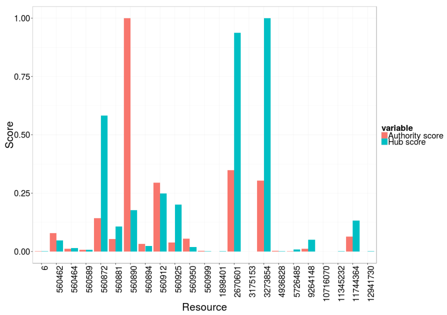
4 Organizational structure

Hub and authority degrees by the Kleinberg algorithm [2] convey a similar meaning to the PageRank. A good hub represents a node that points to many other nodes (*sender*), and a good authority represents a node that is linked by many different hubs (*receiver*). Thus, using this technique, we can identify resources that are important from the perspective of workload distribution in each municipality. The resulting scores are depicted on Figure 5.

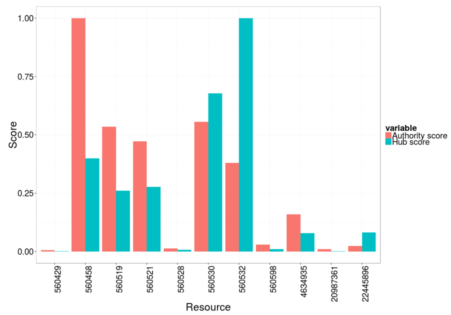
In case of M1 and M3, the individual differences between hub and authority scores support the observations made in Section 3. Namely, out of the three main resources that stand out in M1, one has very high authority score but low hub score, while the other two are high in hub score and low in the authority score. This reflects the clear separation of roles in M1, while also implies a well-established organizational structure. There exist "sender" resources, who delegate or hand the work over to others, as well as "receivers", who receive the work from their colleagues. It is also noteworthy that the "authority resource" executes the administrator role, while the "hubs" perform decision-making activities.

Conversely, in M3 and in M5 the two scores for each individual are almost equal. On the one hand, this might reflect the flexibility of roles in M3, meaning that each resource may appear in different parts of the organization in different cases. On the other hand, it might indicate that the handover of work does not happen from one role to another, but is rather bi-directional.

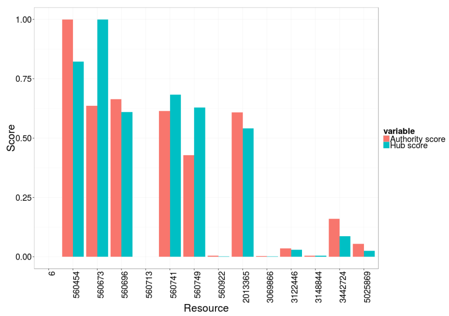
In M4 we notice two main resources who are clearly specialized according to the scores — one is a sender and the other a receiver. The separation is also visible in M2, where we see three main receivers and two senders. Interestingly, in both municipalities all of the mentioned receivers and senders were part of the same clusters in Section 3. Therefore, it is not sufficient to analyse the organizational structure on the role-level. Rather, each resource should be observed individually, as the organization of handover of work can differ within the same role.



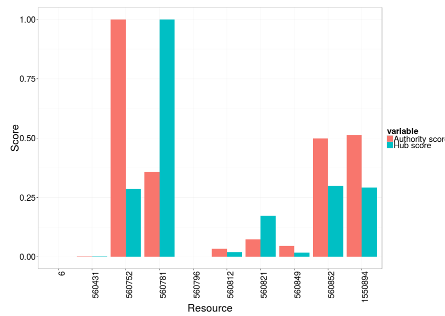
(a) Municipality 1



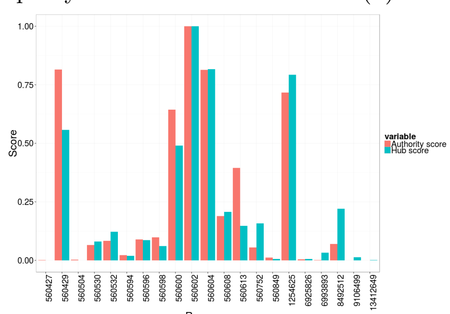
(b) Municipality 2



(c) Municipality 3



(d) Municipality 4



(e) Municipality 5

Fig. 5: Hub and authority scores

We proceed with analyzing the *handover of work networks*, where the nodes represent resources and a directed edge from resource R1 to resource R2 implies that R1 hands the work over to R2. In order to propose relevant improvements on the organizational structure, we need to analyse the up-to-date model. With this purpose, we filter only the cases that were initiated since 2013. This way, we discard the resources who used the information system only a few times when it was established, as well as decrease the effect of resources who have left the work in the municipality.

Even with the time frame filtering, the resulting networks are very "spaghetti-like". This indicates that the organizational structure in all municipalities is rather flexible and in most cases, any resource may directly hand the work over to any other resource. However, these models are hardly readable and, thus, in the following we use a frequency-based filtering of nodes and edges, concentrating the analysis on more frequent behavior.

Figure 6 shows the filtered handover of work network with frequencies, where darker blue color indicates higher frequency. We observe that the case is received by one out of three resources fulfilling the administrator role (560872, 560925, and 560912). The case can be then completed by themselves or delegated to a regular decision-maker (2670601, 3273854, and 11744364) or environmental permit decision-maker (560890 and 560950). Also, the regular decision-makers can hand the work over to an environmental permit decision-maker, who is usually the resource that closes the case.

In general, the administrators do not hand the work over to other administrators, but 560912 can hand it over to 560872 or 9264148. Also, 560912 is often the case-closer.

All of the roles are quite equally represented in the social network of M1. However, one of the administrators, 560872, takes a much higher workload than others (indicated by dark blue color on Figure 6). Similarly, 2670601 takes on much more cases than the other decision-makers. Therefore, one possible improvement could be to manage the workload between the resources performing the same role more evenly.

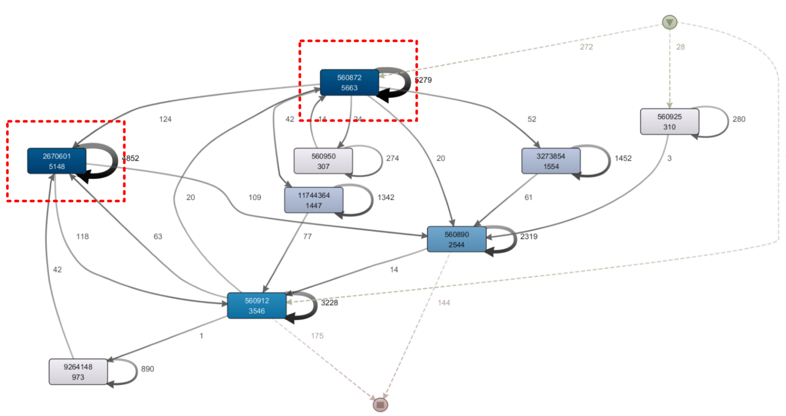


Fig. 6: Handover of work network in M1 (50% of resources, 25% of paths)

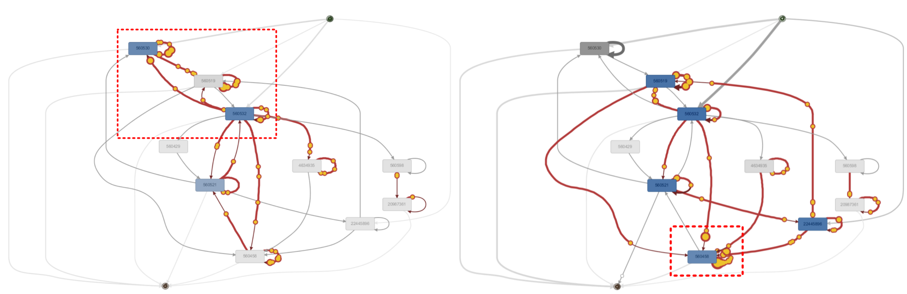
As discovered in Section 3, the roles in M2 are more flexible and most resources are able to perform any activity. Considering this, it would make sense if each resource resolved a case single-handedly from beginning to end, in order to minimize the overhead of handing the case over to other resources.

On the one hand, allowing a resource to take any available task (even if he has not worked on the same case before) helps to optimize the utilization of resources (workloads are more equal). On the other hand, given a case she has not worked on before, the resource needs time to familiarize herself with it. Therefore, we assume that the best way is to divide the cases between currently available resources when the application is submitted and minimize the transfer of the cases afterwards.

Looking at the handover of work network, most resources indeed have very frequently executed self-loops. Three key resources seem to be 560530, 560519, and 560532, who are responsible for the incoming cases. The latter resource also delegates a considerable amount of work to others.

However, we can still find a few points in time when the organization of work is not optimal. Firstly, from the snapshot on Figure 7a we see that the key resources hand some cases over to each other. Assuming that this is not due to the separate responsibilities of these resources, this type of communication could be avoided.

Secondly, resource 560458 seems to be the "case-ending" resource in many occasions. From Figure 7b we see that she can be overloaded with work. Thus, a possible point of improvement could be to delegate some of 560458's responsibilities to another resource.



(a) Snapshot from August 2013 (b) Snapshot from October 2014

Fig. 7: Snapshots of ongoing work in M2, 50% of paths shown

In M3, all of the resources present in the given time frame belong to the same cluster (recall Figure 2c in Section 3). Therefore, their responsibilities should be interchangeable. However, we see a lot of handover of work on Figure 8. If this can not be explained by a hidden separation of duties, then reducing the handover of work could improve the process. On the positive side, all resources in this municipality seem to be actively participating.

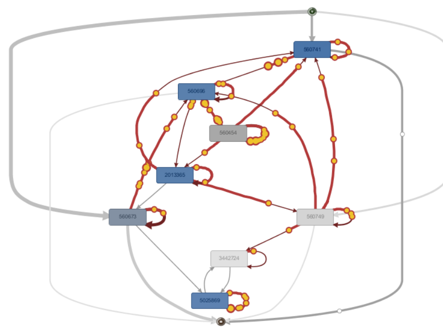


Fig. 8: Snapshot of ongoing work in M3 from August 2014, 50% of paths shown

In case of M4 we encounter again the separation of duties between two roles. In the given period, there are two main active administrators (560781 and 560752) and two main active decision-makers (1550894 and 580852). Figure 9 illustrates that 560752 can be overloaded with work.

This issue can be further investigated by looking at the resource-phase combinations on Figure 10. The application is received by either of the two administrators, then delegated to either of the two decision-makers (or completed by 560781 itself). After the application is declared receptive and advice is known, the case goes back to the administrators (usually 560781) for marking the decision as taken and sending it. Finally, the case is handed to the overloaded 560752 who then claims the decision irrevokable.

Three resources in this municipality appear to be rather inactive (not counting resource 6 who is an infrequent cross-municipal resource as seen in Table 3). It is unclear whether these resources are simply not supposed to actively participate in the process or if they are left out due to a bad organization of workload.

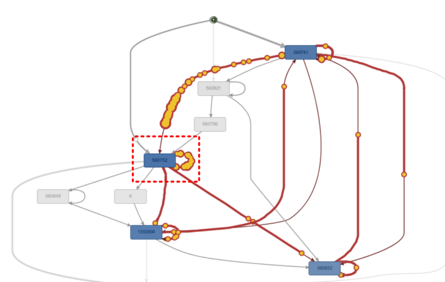


Fig. 9: Snapshot from February 2014, 50% of paths shown

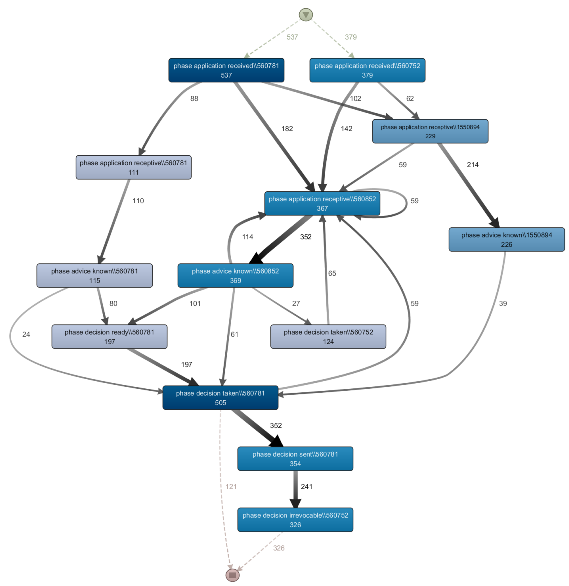


Fig. 10: Process model with resource-phase combinations as nodes, 5% of nodes and 5% of paths shown

In M5, we see two resources with high incoming degree: 560600 and 560602. These two resources belong to different clusters. Resource 560600 can be characterized by performing Application receptive and Advice known, while resource 560602 by performing Decision sent. Resource 560600 is often overloaded with work (see Figure 11).

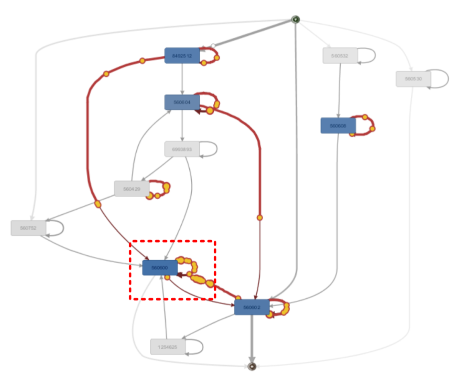


Fig. 11: Snapshot of ongoing work in M5 from April 2014, 50% of resources and 25% of paths shown

5 Process drift

As it was mentioned earlier, two municipalities have moved to the same location at some point in time. Our findings from Section 3 imply that two resources from M2 moved to the same location as M5 in May 2014. However, this is only an assumption and it is not conclusively known which of the municipalities are involved or when the location merge has happened. Therefore, in order to estimate the effect of this particular change, we first have to identify the potential locations of the change in the processes of each municipality and compare them.

We address aforementioned problems by performing *changepoint analysis* (also called *drift detection*). Detection of changepoints is a challenging task, and many attempts have been made to tackle it – from simple CUSUM-based charts [3] to advanced non-parametric methods [4]. We adopt and compare two methods: a novel method, called ProDrift, which is designed specifically for the business process drift detection [5] and the binary segmentation algorithm [6], one of the widely applied methods for multiple changepoints detection. In a nutshell, the ProDrift method takes the whole log as an input, transforms traces into partial order runs and applies Chi-square test on a sliding window.

The binary segmentation algorithm, on the other hand, is a generic method that applies on univariate timeseries and detects multiple structural changes using mean, variance, or both. We use the binary segmentation on two different monthly timeseries; *trace timeseries* represent monthly aggregates of total number of different traces that were finished in this month, whereas *event timeseries* are median numbers of unique events in these traces.

As our goal is to detect a structural shift, either in a municipality workload or a variety of the procedures, we do not want the result to be affected by the seasonal/periodic effects. Therefore, before applying the method, we decompose it to the trend, periodic component and the remaining part. The decomposition of the timeseries is itself a very powerful method, which sheds the light on different aspects. For example, from the figure of trace timeseries decomposition (Figure 12) we may notice that M4 has the strongest seasonal component with the higher peaks of trace numbers closed each June and April and decrease of workload on August, whereas total of finished traces of M2 peaks on January. In general, it is notable that all municipalities have very different periodic components that do not overlap. In terms of variety of procedures the periodic component is less expressed (Figure 13), but also does not have any overlapping periods.

Second important aspect is the trend of the timeseries (Figure 12), which in the case of the traces is very stable for all the municipalities across the years. We note that M3 is somehow different from other municipalities, but it can hardly be called a drastical change. However, trends of events (Figure 13) have higher differences in the trends. M2 and M5 have initially very similar process variety, but in 2013-02/2013-03 M2 increases in the median of performed unique events, while M5 decreases on almost the same amount. Earlier, we discovered that cross-municipality resources tend to switch their work from M2 to M5 and vice versa (Figure 4), which contributes to the hypothesis that M2 and M5 might be the ones who moved into the same building. However, M1, M3, M4 represent another cluster, where they have very similar trend until January 2013, but later on, M1 exhibits increasing trend, while other two decrease.

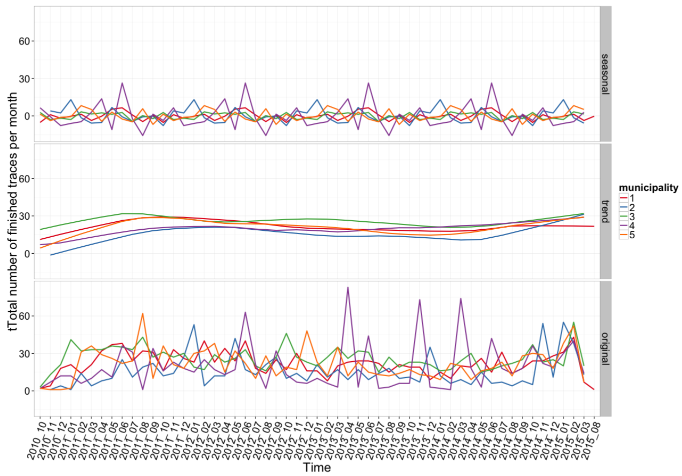


Fig. 12: Comparison of the decomposed seasonal component and the trend of **trace number** between municipalities

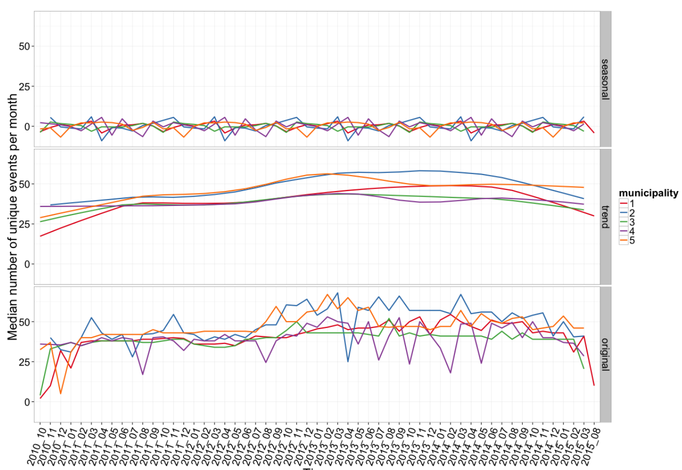


Fig. 13: Comparison of the decomposed seasonal component and the trend of **events** between municipalities

As we applied additive decomposition, we simply subtract seasonal component from the data and detect both mean- and variance-based changepoints on unseasoned timeseries. The results are shown for traces on Figure 14 (for changepoints of events see Figure 22 in Appendix). Unfortunately, the detected periods (red lines correspond to the periods between changepoints), despite being visually justifiable, do not provide us with clear answer, which municipalities have been merged. Therefore, we decided to compare different changepoints from different methods and against real changes in regulations taken from the municipality page[1]. On Figure 15 we listed all possible dates, where `real_events` indicate some changes according to the municipality page, `trace_M(1:5)` - detected dates of changes in the deseasoned trace timeseries for each municipality, whereas `event_M(1:5)` – the same information for the event timeseries. Last, but not the least is the ProDrift method.

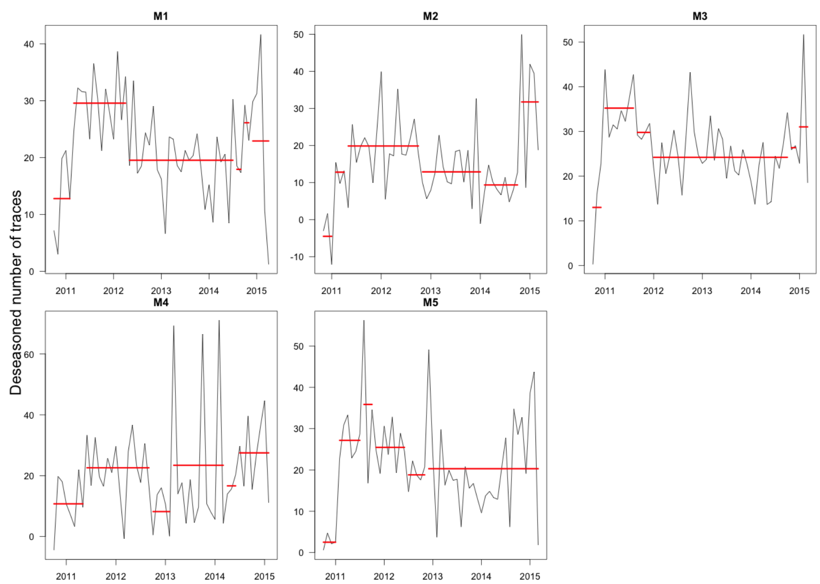


Fig. 14: Changepoint detection on deseasoned timeseries of traces

[1] `http://www.infomil.nl/onderwerpen/ruimte/mer/overzicht/`

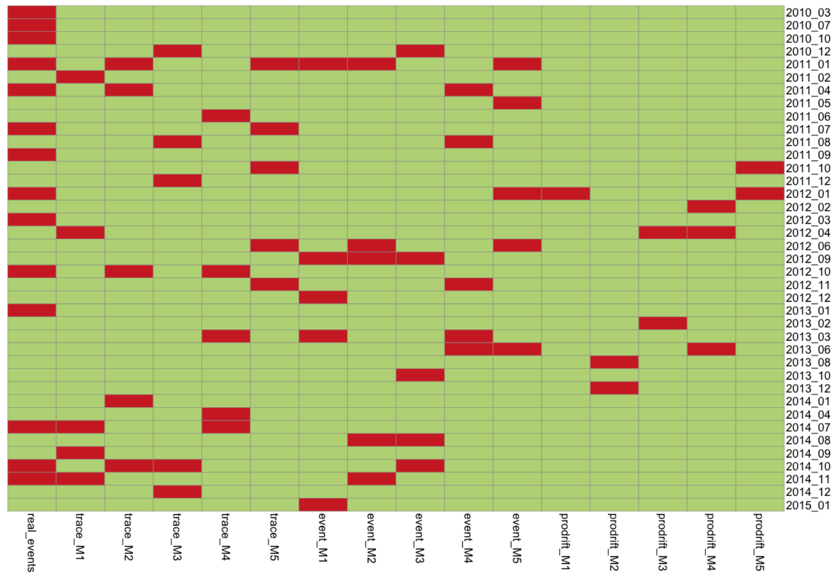


Fig. 15: Comparison of detected changepoints

From the results we observe that some real events (e.g. July 2010 "The Modernization Act of the regulations") are not detected by any method, while others supposedly affected multiple municipalities (January 2011 "Second tranche Activities Decree" was detected as a change in M2, M1, M5 in traces and events timeseries). One should be very careful interpreting these results, as these changes can be caused by other factors, as well as the real events can affect the results later in time.

However, it is more interesting to check changes that do not correspond to the real events, but different methods still agree about the drift. For example, 2013-03 and 2013-06 correspond to a relatively short interval, when changes have happened according to the different methods in M1, M4, and M5. It can also potentially indicate that some common event caused changes in these municipalities.

In general, the changepoints found with the three methods do not coincide. Especially, the ProDrift tool gives rather different results from the other two methods. For example, according to the ProDrift results there was a major change in early 2012 that affected all municipalities except M2.

Still, it is difficult to claim, what municipalities physically moved to the same location. The results do not support our hypothesis about M2 moving to the same location as M5 in the spring of 2014. Therefore, either the methods are not able to detect the effect of this change or the merge did not cause drastical changes in the number of traces or event variety.

6 What-if analysis

In order to investigate the effect of outsourcing various tasks, we create a simulation. First, we need to construct an "as-is" model that reflects the current behavior. Then, we can eliminate some parts of the process and see how the process is affected by the change. The BIMP simulator[2] is used for running the simulations.

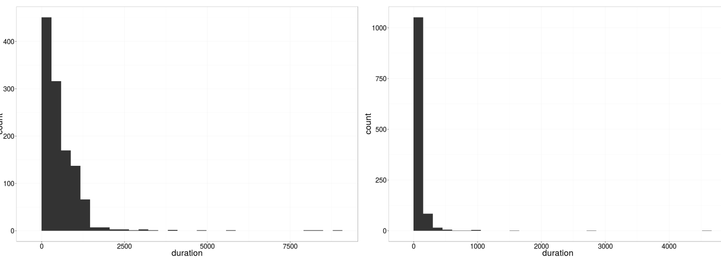
We decided to analyse the effects of outsourcing on the process as a whole, merged from the 5 municipalities. This way, we can quickly estimate the major effects of outsourcing different parts of the process. Later, the same analysis can be redone on chosen parts for each municipality separately to investigate the effects that are specific to the municipalities.

In order to obtain the as-is model, we constructed a merged log out of the separate logs of the 5 municipalities. For capturing the up-do-date behavior, we filtered only cases which are contained in the time frame between January 1, 2013 and December 31, 2014. With the purpose of describing the most frequent behavior, we used the phase-level activities and filtered only the cases that belong to the most frequent variants (occurring at least 25 times over the whole data period). As a result, we were left with a log with 1351 cases in 18 variants.

We proceeded to set up the simulation in the following steps.

1. Built a process model from the merged logs. The Fuzzy model of the process can be seen on Figure 23 in Appendix. In BIMP simulator, a corresponding BPMN model is used.
2. Added branching probabilities to the XOR-splits in the BPMN model. The probabilities were estimated based on the frequencies on the Fuzzy model (Figure 23).
3. Set the duration distribution for each phase. We found that the durations for each phase follow an exponential distribution (see examples on Figures 16a and 16b). The mean duration was used as a distribution parameter in BIMP. The mean durations for each phase can be seen in Table 4.
4. Each phase was matched with a role that performs it. For each phase, we considered the dominant resource (the resource with highest frequency of performing this phase) and identified its role as either administrator or decision-maker based on the clusters on Figure 2. The roles are brought in Table 4.
5. The number of administrators was set to 15 and number of decision-makers to 31, following the actual numbers in the merged log.
6. The inter arrival time of cases was set to 13 hours, computed as an average from the merged log.

[2] `http://bimp.cs.ut.ee`



(a) Application receptive (b) Decision taken

Fig. 16: Histograms of durations for two sample phases

Table 4: Phase durations and performing roles for simulation

Phase	Role	Mean duration (hours)
Decision irrevocable	Administrator	982.356
Application received	Administrator	88.536
End	Administrator	68.548
Decision sent	Administrator	62.512
Decision taken	Administrator	53.156
Draft decision made	Administrator	4.000
Procedure aborted	Administrator	0.006
Archived case	Decision-maker	1,403.438
Permanently suspended	Decision-maker	871.624
Asked additional information	Decision-maker	759.022
Concept draft decision ready	Decision-maker	753.279
Procedure prematurely terminated	Decision-maker	684.699
Application receptive	Decision-maker	536.760
Additional information received	Decision-maker	255.366
Decision ready	Decision-maker	105.430
Concept decision ready	Decision-maker	64.952
Draft decision ready	Decision-maker	47.889
Advice known	Decision-maker	26.873
Case handled	Decision-maker	0

When constructing what-if scenarios, we need to keep in mind that some procedures can not conceptually be outsourced at all, such as Decision taken or Case handled. For one possible scenario, we can assume that the initial check of the application could be done before the submission, making phases *Asked additional information*, *Additional information received*, and *Procedure permanently suspended* redundant.

In order to simulate this scenario, we removed these three activities from the process model and adjusted the branching probabilities accordingly (see the corresponding BPMN model on Figure 24 in Appendix). We ran 3 simulations with 2000 cases for both the as-is process and the constructed scenario. The mean results of the three runs are presented in Table 5. The time estimates (in hours, weeks) in the simulation seem quite unreliable, so we will use them only as abstract *time units* for comparing the two scenarios.

The total duration of completing 2000 cases is about 25% smaller in the what-if scenario (Table 5). The percent of resource utilization shows how much of the time the resources of a given role were occupied with work. The higher the utilization score, the better the resources are organized in terms of work. The utilization of administrators is higher in the as-is scenario, while the utilization of decision-makers is better in the what-if scenario. Overall, the resources are utilized better in the what-if scenario. Thus, we can conclude that outsourcing application check lowers the duration of the process and results in a better utilization of existing resources.

Table 5: Simulation results

	Total duration	Administrator util.	Decision-maker util.
As-is	6249.6 time units	87.6%	65.7%
Application checked	4720.8 time units	81.3%	79.7%

Figure 17 shows the average waiting times for each phase, again measured in abstract time units. For several phases, the waiting time actually increased slightly in the what-if scenario. However, when the waiting time decreased, it did so significantly. For example, the waiting times of Decision irrevokable, Decision sent and Draft decision made have decreased by half or more. In total, the waiting times would be lower if the application check was outsourced.

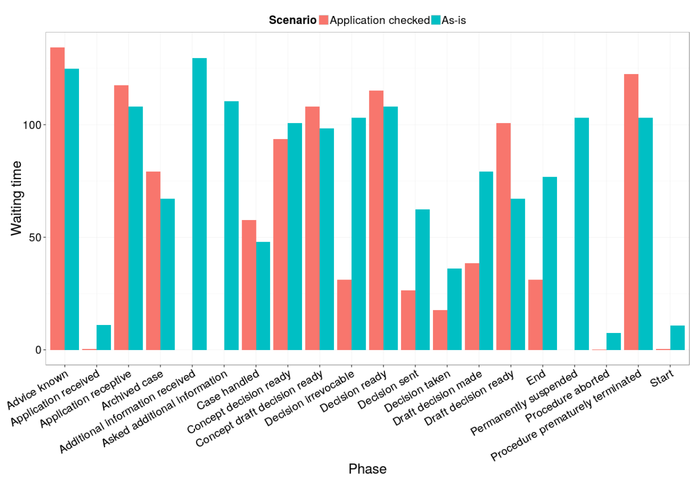


Fig. 17: Average waiting times according to the simulation

However, a simulation has several drawbacks. Firstly, it requires lots of assumptions. For example, the assumption that the Decision taken phase is performed only by administrators or decision-makers is not valid, because actually this phase can be performed by both roles. Thus, the resulting model deviates from the reality. Secondly, the durations for each phase have very high variability, which makes the simulation unstable. Running the simulation with the same setting multiple times can result in very different outcomes.

A relatively good indication of what will happen to the process can be obtained by simply looking at Table 4. The higher the duration of the outsourced phase, the more it affects the overall duration of the case, as well as the utilization of resources. The dominant role gives a hint about which role will be affected more by the change.

7 Performance analysis

In this section, we address the question about throughput times and their differences between municipalities. As it was mentioned earlier in Section 2, we obtained the duration between pairs of events. We adopt the same notion in order to perform performance analysis and investigate how median durations (as a more robust measure compared to mean) of the same pairs of activities differ across municipalities. As the whole unique set of event pairs is very large ($N = 5097$), we will focus on:

- event pairs with median duration larger than 0.1 seconds ($N = 3301$)
- event pairs that were performed in all municipalities ($N = 210$)

We realize that this way we lose information regarding some events (e.g. where usually it takes 0 sec, but some municipalities perform longer or only some municipalities are performing these activities). However, we believe that we are left with interesting insights. On Figure 18 we observe 50 event pairs with highest variability and their median durations for each of the municipalities. The rest of the event pairs are presented in Appendix (Figures 25 and 26).

Most of the event pairs take little time to perform and have little or no variation between the municipalities. However, some durations of event pairs have particular "outlier" in terms of the taken time. In such cases, the reduction of throughput times should be possible, because we know that in other municipalities the same event pair takes less time.

Although M1 is rarely seen as an outlier in terms of longer times, it happens mostly when one or both of the events are related to entering or registering the date for some procedure (Figure 18). For M2, various event pairs appear as time-consuming. According to such terms as "objection lodged", "term extension", "complete completeness", as well as "additional data" and "additional information", M2 handles longer than others cases that are extended or otherwise delayed. M3 has the highest delay in event pair *enter senddate procedure confirmation-register submission date request*, which is related to the beginning of the process. M4 experiences very high delays in multiple event pairs, such

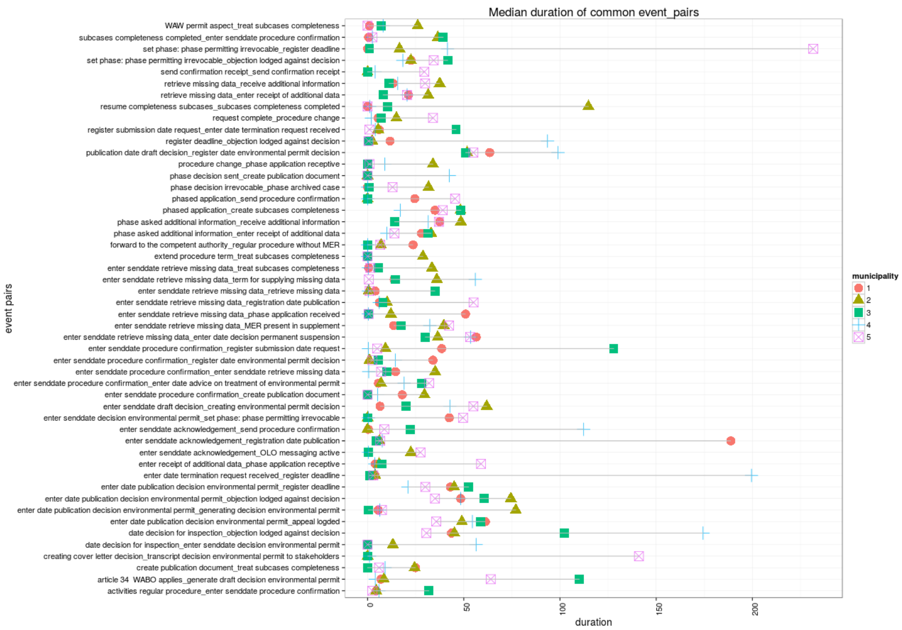


Fig. 18: Median duration of event pairs in municipalities, part I

as *enter date termination request received-register deadline* and *enter senddate acknowledgment-send procedure confirmation*. M5 as a municipality seems to have longer times for event pairs like *set phase: phase permitting irrevocable-register deadline* or *creating cover letter decision-transcript decision environmental permit to stakeholders*.

This type of performance analysis is a valuable tool for the domain knowledge specialists to retrieve information about the differences in throughput times. Based on our observations, the processes are not identical in the five municipalities and there exist several points for improvement. Focusing on the bottlenecks highlighted by our analysis, the throughput times could be significantly reduced.

8 Control flow

In order to estimate how similar the five municipalities are from control flow perspective, we started with conformance checking on the phase-level logs. More precisely, we discovered process models (using Heuristic miner in ProM) for each municipality and replayed the event logs of all other municipalities on the discovered models. This technique did not imply significant differences between the municipalities; the resulting fitness scores were between 0.84 and 0.98 in all cases (see Table 6).

Table 6: Fitness between the event log and process model

	Log				
Model	M1	M2	M3	M4	M5
M1	0.972	0.918	0.974	0.948	0.945
M2	0.843	0.970	0.864	0.847	0.857
M3	0.933	0.900	0.936	0.944	0.905
M4	0.967	0.931	0.972	0.979	0.944
M5	0.916	0.919	0.923	0.909	0.933

Therefore, we proceed with more granular analysis by calculating the frequencies of low-level activities in each municipality. The frequencies are normalized by the total number of events in the corresponding municipality. Next, we find the top 10 activities where the difference between the maximum frequency and the mean frequency in other municipalities is maximal.

The frequencies of these activities are plotted on Figure 19. One thing to note here is that M1, M3, and M5 often follow similar patterns, having higher frequency in the same activities, especially *enter senddate decision environmental permit* and *send confirmation receipt*. Furthermore, when M2 has high frequency M5 is also high, namely, *close case* and *set phase: phase permitting irrevocable*.

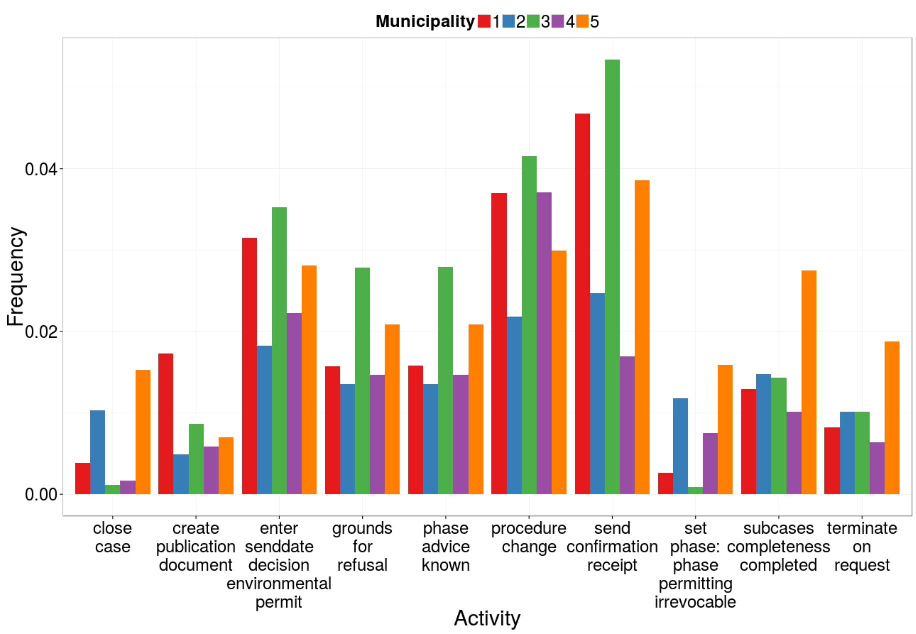
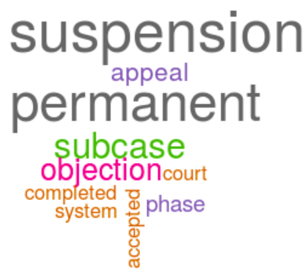


Fig. 19: Top 10 activities that differ in terms of frequency between the municipalities

Still, each municipality has its own niche as well. M1 stands out in *create publication document*. M3 has the highest frequency in several activities, e.g. *grounds for refusal* and *phase advice known*. M4 is always dominated by other municipalities, but still, its high frequency of *procedure change* is noteworthy. M5 is much more frequent in performing *subcases completeness completed* and *terminate on request*.

Going even further, we investigate which activities occur only in one of the municipalities. Word clouds of corresponding events can be seen on Figure 20. An interesting thing to observe is that in M2 there are some events related to *objection*, *court*, and *appeal*, while in M5 we see *court* and *hearing*. M2 has 3 unique phase-activities: *phase further information requested*, *phase further information received*, and *phase appeal to higher court*. Moreover, there are no activities specific to M4. The frequency of all municipality-specific events is quite low, so we should be cautious in making conclusions here.

(a) M1-specific activities



(b) M2-specific activities

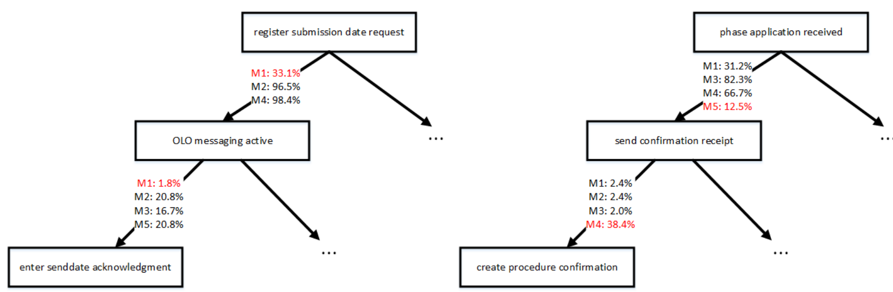
(c) M3-specific activities

(d) M5-specific activities

Fig. 20: Municipality-specific activities

Additionally, we discovered differences in control flow based on *event structures*. The method of van Beest et al. [7] takes two event logs, encodes them as event structures, and identifies differences in those structures. The differences are presented as sentences in natural language. The method is implemented as a tool called ProDelta[3].

For performance reasons, we compare only the logparts taken after the last changepoint in each municipality according to drift detection by ProDrift as discussed in Section 5. An example output sentence obtained when comparing M1 and M2 is: *In model 1, after the execution of the 1st occurrence of "register submission date request" the frequency of branching to the 1st occurrence of "OLO messaging active" is 33.1%; whereas in model 2, after the execution of the 1st occurrence of "register submission date request" the frequency of branching to the 1st occurrence of "OLO messaging active" is 96.5%*. Figure 21 illustrates some examples of the most significant results in two sketches.



(a) (b)

Fig. 21: Sketches for branching points

Based on the observations we can conclude that noteworthy differences happen already at the beginning of the process. *Register submission date request* is the most common start event in all municipalities; and is often followed by *OLO messaging active*. M1 stands out at the earliest phase of the process, having lower frequency for this branching pattern (Figure 21a). However, this does not mean that OLO messaging active is performed in less cases in M1. Rather, there exist intermediate activities that can be performed between those two activities. Similarly, branching from *OLO messaging active* to *enter senddate acknowledgement* has lower frequency in M1 than in other municipalities.

The activities involved in the second sketch also happen early in the process. *Phase application received* marks the end of the first phase and often follows directly OLO messaging active. In the first branching point, M5 is the outlier, having the lowest frequency (Figure 21b). Also, M1 is notably lower that others in this point. In the second branching point on this sketch, M4 stands out as having much higher frequency than others in branching from *send confirmation receipt* to *create procedure confirmation*.

[3] Available at `http://apromore.org/platform/tools`

9 Conclusion

In this work, we analyzed building permit application process in five Dutch municipalities. We investigated the process from organizational, time, and control flow perspectives. Below we will summarize our work by giving short answers to the 6 research questions that guided our analysis.

> 1. What are the roles of the people involved in the various stages of the process and how do these roles differ across municipalities?

We were able to identify two main roles in the municipalities: administrators and decision-makers. Administrators receive the application and in simple cases resolve them. If necessary, administrators delegate the case to decision-makers who collect advice and make the decision based on that. The separation of duties is most clearly visible in M1, while in M2 and M3 the resources are usually able to perform both roles.

> 2. What are the possible points for improvement on the organizational structure for each of the municipalities?

Possible improvements on the organizational structure can be divided into two main directions. For M2 and M3, where the separation of duties does not clearly emerge, the unnecessary handover of work between resources could be reduced. If any resource is able to perform any activity, it would be simpler if the same resource would work in a case from beginning to end. In M1, M2, M4, and M5 we were able to detect specific resources who are overloaded with work. As the tasks are more clearly divided between the roles in these three municipalities, possible solutions would be to either hire new people with the same role as the overloaded resource or to reorganize the responsibilities between the existing resources.

> 3. The employees of two of the five municipalities have physically moved into the same location recently. Did this lead to a change in the processes and if so, what is different?

According to our assumption, two resources from M2 moved to the same location with M5 in May 2014. However, none of the changepoint detection methods we used were able to detect a change in this period. We find that the absence of a definitive answer about which municipalities were involved in the workplace change indicates that the changes were not significant from the perspective of traces and events. Still, we identified other possible changepoints in process structure and compared the results against modifications in regulations. Furthermore, our findings imply that there were changes that affected most of the municipalities in the early 2012, as well as in the spring of 2013.

> 4. Some of the procedures will be outsourced from 2018, i.e. they will be removed from the process and the applicant needs to have these activities performed by an external party before submitting the application. What

will be the effect of this on the organizational structures in the five municipalities?

We provided a simulation framework for analyzing the effects of eliminating some parts of the process. We illustrated the simulation results on a sample scenario — outsourcing the initial check of the application. According to the simulation, the process time in the scenario is reduced by about 25% as compared to the as-is process and the utilization of resources is more optimal.

5. Where are differences in throughput times between the municipalities and how can these be explained?

For each municipality we identified event pairs that differ in median duration as compared to other municipalities. In case of M1, those event pairs tend to be related to entering or registering dates for some procedure. One general explanation for longer delays when entering dates could be that these events are often the first or last events in larger phases and, thus, these durations imply the transfer from one phase to another. M2 stood out with higher delays in extended or delayed cases. M3 had higher duration of performing some of the starting activities. Overall, M5 and M4 were the most frequent outliers, handling multiple event pairs longer than others.

6. What are the differences in control flow between the municipalities?

The five municipalities are relatively similar from control flow perspective in phase-level activities. Still, there exist differences in the lower level. In terms of frequencies of performing single events, we found that M1, M3, and M5 often follow similar patterns, having higher frequencies than M2 and M4. Also, M2 seems to be connected to M5, having no distinctive differences of its own. M4 did not stand out as a single outlier in any activities. Differences in branching from one activity to another mostly happen at the beginning of the process, where M1 was identified as the biggest outlier.

Appendix

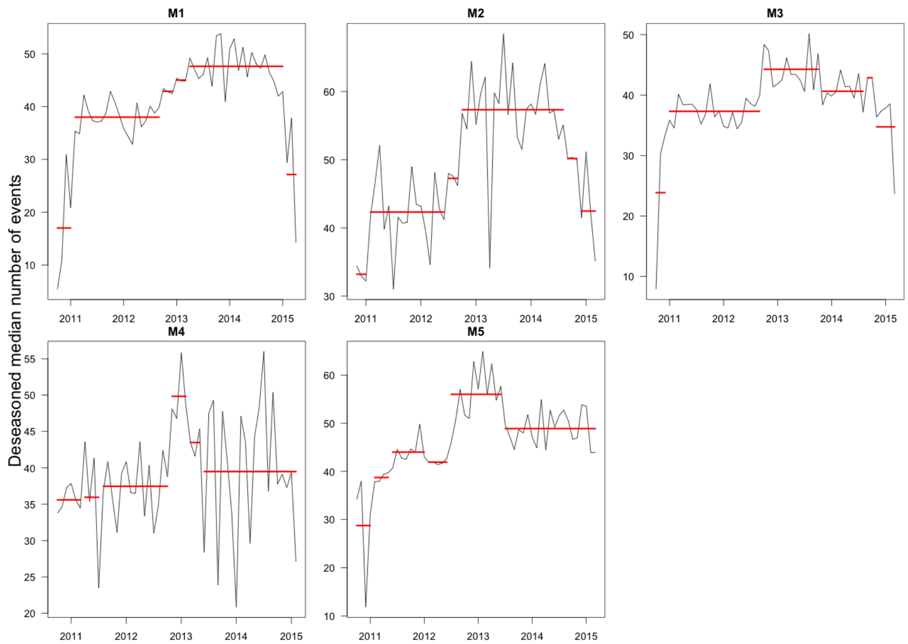


Fig. 22: Changepoint detection on deseasoned timeseries of events

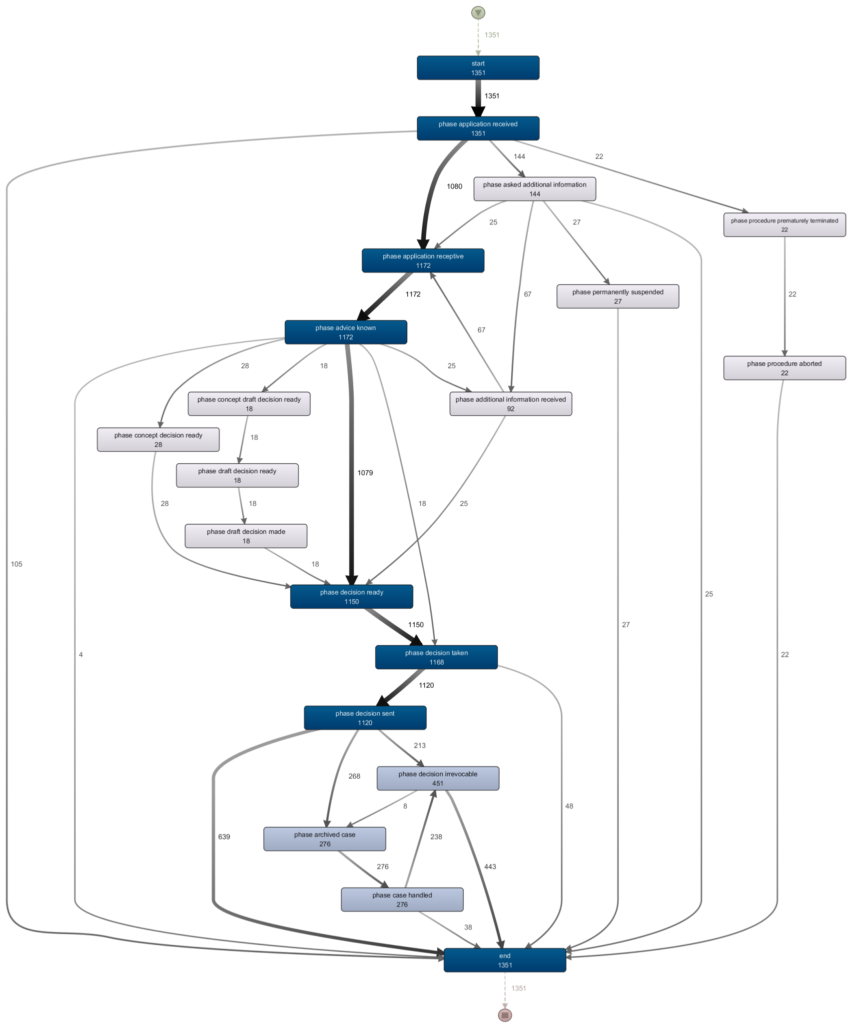


Fig. 23: Process model for the merged logs

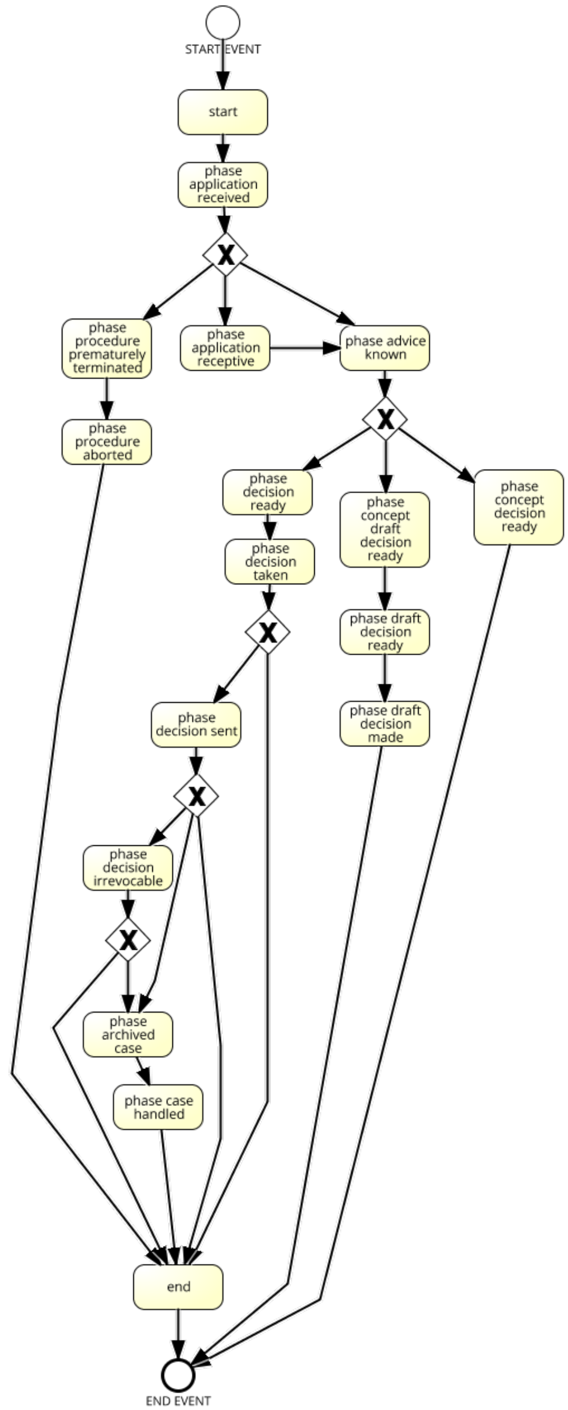


Fig. 24: BPMN model for the what-if scenario

Fig. 25: Median duration of event pairs in municipalities, part II

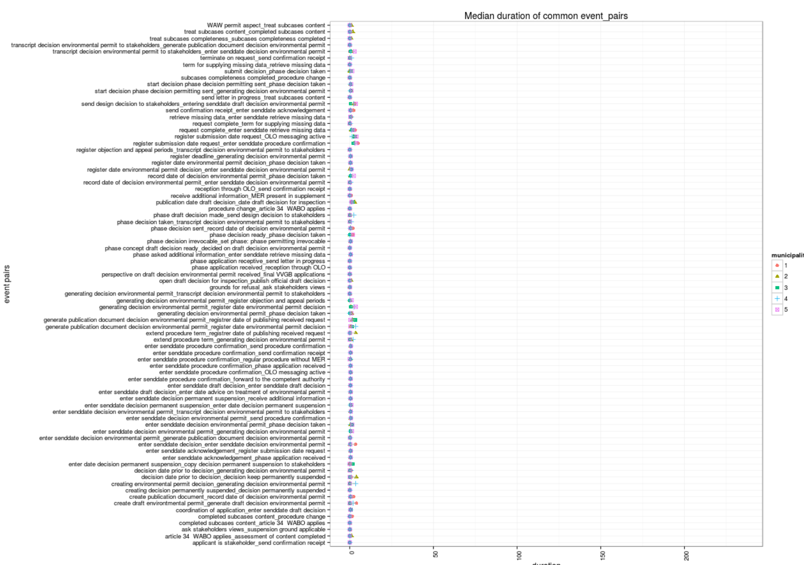


Fig. 26: Median duration of event pairs in municipalities, part III

References

1. Fraley, C., Raftery, A.E.: Enhanced model-based clustering, density estimation, and discriminant analysis software: MCLUST. Journal of Classification **20** (September 2003) 263–286
2. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) **46**(5) (1999) 604–632
3. Page, E.: Continuous inspection schemes. Biometrika (1954) 100–115
4. James, N.A., Matteson, D.S.: ecp: An r package for nonparametric multiple change point analysis of multivariate data. arXiv preprint arXiv:1309.3295 (2013)
5. Abderrahmane, M., Dumas, M., La Rosa, M., Ostovar, A.: Fast and accurate business process drift detection. (2015)
6. Scott, A., Knott, M.: A cluster analysis method for grouping means in the analysis of variance. Biometrics (1974) 507–512
7. van Beest, N.R., Dumas, M., García-Bañuelos, L., La Rosa, M.: Log delta analysis: Interpretable differencing of business process event logs. (2015)