Dealing With Concept Drifts in Process Mining: A Case Study in a Dutch Municipality

R.P. Jagadeesh Chandra Bose, Wil M.P. van der Aalst, I. Žliobaitė and M. Pechenizkiy

Department of Mathematics and Computer Science,
Eindhoven University of Technology, The Netherlands.

Abstract. Although most business processes change over time, contemporary *process mining* techniques tend to analyze these processes as if they are in steady-state. Processes may change suddenly or gradually. The drift may be periodic (e.g. due to seasonal influences) or one-of-a-kind (e.g., the effects of new legislation). For process management it is crucial to discover and understand such *concept drifts* in processes. In this paper, we present a case study of analyzing concept drifts in three different processes within a large Dutch municipality.

1 Introduction

In recent years process mining techniques have matured. Provided that the process is stable and enough example traces have been recorded in the event log, it is possible to discover a high-quality process model that can be used for performance analysis, compliance checking, and prediction. Unfortunately, most processes are not in steady-state. In today's dynamic marketplace, it is increasingly necessary for enterprises to streamline their processes so as to reduce costs and to improve performance. Moreover, today's customers expect organizations to be flexible and adapt to changing circumstances. New legislations such as the WABO act [1] and the Sarbanes-Oxley Act [2], extreme variations in supply and demand, seasonal effects, natural calamities and disasters, deadline escalations [3], etc., are also forcing organizations to change their processes. For example, governmental and insurance organizations reduce the fraction of cases being checked when there is too much work in the pipeline. In case of a disaster, hospitals and banks change their operating procedures etc. It is evident that the economic success of an organization is more and more dependent on its ability to react and adapt to changes in its operating environment. *Concept drift* refers to the situation in which the process is changing while being analyzed. There is a need for techniques that deal with such *second order dynamics*. Analyzing such changes is of utmost importance when supporting or improving operational processes and to get an accurate insight on process executions at any instant of time.

The remainder of this paper is organized as follows. Section 2 provides the background on change detection techniques based on hypothesis tests. The case study of analyzing concept drifts in three processes of a large Dutch municipality is presented in Section 3. Section 4 concludes the paper.

2 Background

Processes can change in with respect to the three main process perspectives, viz., control-flow, data, and resource. Such changes are perceived to induce a drift in the concept (process behavior), e.g., in the way which activities are executed *when*, *how*, and by *whom*. There are three topics when dealing with concept drifts in process mining.

1. *Change Point Detection:* The first and most fundamental problem is to detect concept drift in processes, i.e., to detect that a process change has taken place. If so, the next step is to identify the time periods at which changes have taken place. For example, by analyzing an event log from an organization (deploying seasonal processes), one should be able to detect that process changes happen and that the changes happen at the onset of a season.
2. *Change Localization and Characterization:* Once a point of change has been identified, the next step is to characterize the nature of change, and identify the region(s) of change (localization) in a process. Uncovering the nature of change is a challenging problem that involves both the identification of change perspective (e.g., control-flow, data, resource, sudden, gradual, etc.) and the identification of the exact change itself. For example, in the example of a seasonal process, the change could be that more resources are deployed or that special offers are provided during holiday seasons.
3. *Change Process Discovery:* Having identified, localized, and characterized the changes, it is necessary to put all of these in perspective. There is a need for techniques/tools that exploit and relate these discoveries. Unraveling the evolution of a process should result in the discovery of the change process describing the second order dynamics. For example, in the example of a seasonal process, one could identify that the process recurs every season. Also, one can show an animation on how the process evolved over a period of time with annotations showing several perspectives such as the performance metrics (service levels, throughput time, etc.) of a process at different instances of time.

One can consider an event log $\mathcal{L}$ as a time series of traces (traces ordered based on the timestamp of the first event). The basic premise in handling concept drifts is that *the characteristics of the traces before the change point differ from the characteristics of the traces after the change point.* The problem of change (point) detection is then to identify the points in time when the process has changed, if any. Change point detection involves two primary steps: (i) capturing the characteristics of the traces, and (ii) identifying when these characteristics change.

The control-flow perspective of a process characterizes the relationships between activities. Dependencies between activities in an event log can be captured and expressed using the *follows* (or *precedes*) relationship, also referred to as *causal footprints.* Bose et al. [4] proposed four features characterizing the control-flow dependencies between activities. These features are shown to be effective in detecting process changes. An event log can be transformed into a data set $\mathcal{D}$, which can be considered as a time series (as depicted in Fig. 1), by these features. Change detection is done by considering a series of successive populations[1] of feature values (of some population size w, see Fig. 1) and investigating if there is a significant difference between two successive populations. The premise is that differences are expected to be perceived at change points provided appropriate characteristics of the change are captured as features. The difference between populations is assessed using *statistical hypothesis testing* [5]. Hypothesis tests yield a significance value (the so-called *p-value*), whose range is between 0 and 1, assessing the validity of the null-hypothesis, which typically states that the two populations come from the same distribution. A plot of p-values corresponding to the trace indices captured by populations is inspected to see if significant differences (and thereby process changes) exist. *The p-values are plotted against the indices at the end of the left populations.* Fig. 2 depicts a representative p-value plot. Process changes stand out as *troughs* in the p-value plot. This approach is effective in detecting sudden drifts as shown in [4, 6].

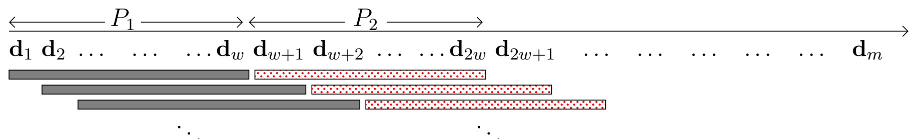


Fig. 1. Basic idea of detecting drifts using hypothesis tests. The data set of feature values is considered as a time series for hypothesis tests. P_1 and P_2 are two populations of size w.

Techniques for dealing with concept drift can be broadly classified into *online* and *offline* depending on whether or not the presence of changes or the occurrence of drifts needs to be uncovered in real-time.

3 Case Study in a Dutch Municipality

In this section we present a case study of analyzing concept drifts in three processes within a large Dutch municipality. Municipalities are interested in getting insights into their processes, e.g., the way they are planned to be executed vis-a-vis the way they are actually executed. Moreover, they want to know which parts/regions in processes are time consuming. Municipalities find such insights important and interesting for many reasons. For example, in some cases, municipalities can only charge its customers based on the real costs for providing a

[1] A moving window is used to generate the series of populations.

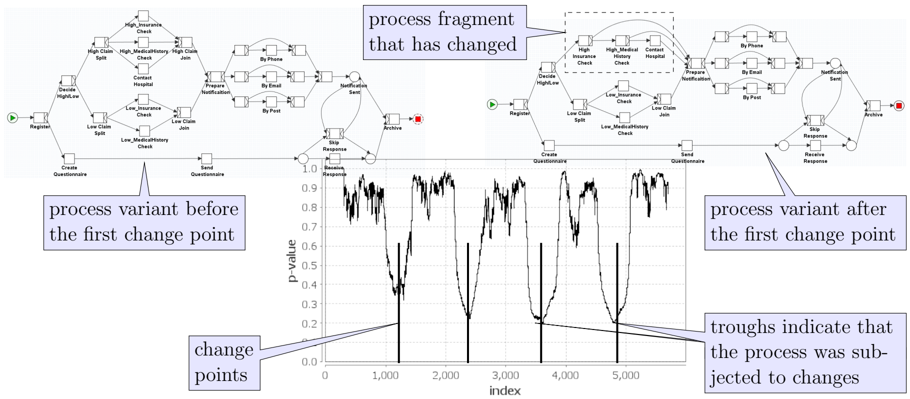


Fig. 2. A plot of p-values of the hypothesis tests. X-axis represents the trace index and Y-axis represents the p-value. Troughs in the plot signify process changes. Process variants before and after a change point can be inspected to identify the fragments that have been changed.

service and in other cases they can charge a fixed fee for a service. Also, in case of permit requests, if the municipalities do not come to a decision within a certain time (as set by the law) then the permit has to be granted. Therefore, they want to be as cost efficient as possible. Furthermore, processes within different municipalities are very similar in many aspects. At the same time, each municipality can have its own characteristics (e.g., differences in size, demographics, problems, and policies) that need to be maintained. Recently, different municipalities in the Netherlands have evinced interest in comparing their processes and learning from each other (the interested reader is referred to the CoSeLog project [7] for further information.). Their vision is to have a form of standardization through a centrally managed process management system [8–11]. This includes the definition of configurable process models allowing for variations peculiar to each municipality. The configurable process model can be realized from a set of concrete models (normative models) that capture the desired or required behavior. However, more often than not, the concrete models are either not available or even if available are of very low quality. Process mining plays a significant role in bridging this gap by enabling the discovery of what the actual processes are. One can consider both normative models as well as discovered models in extracting configurable process models. This makes the discovery of good and correct process models (reflecting the reality) from event logs extremely crucial. When analyzing event logs, one needs to factor in the possibility of process changes, i.e., concept drifts, that could have taken place. In the following sections, we present the results of analysis of concept drifts in event logs pertaining to three processes from this municipality related to building permits.

3.1 Permit Process for Advertisements

In this section, we present the results of analysis of concept drifts in event logs pertaining to one of the processes related to permits for advertisements. If a person/organization wants to advertise on a building in the Netherlands, for example on a billboard or an illuminated sign, a permit is needed in most cases, which can be obtained from the local municipality. The municipality may charge an advertising tax or municipal tax on encroachments on or above public space (precariorechten) for advertisements visible from the public road.

We considered an event log containing 116 cases and 2335 events referring to 25 activities. The cases pertain to permit requests for placing advertisements spanning over the period between 07-07-2003 and 18-03-2008. We considered the J-measure feature on the *follows* relation for all activity pairs using a window of size 10. The choice of window size of 10 was made based on the characteristics of the process. The process has four high-level sub-processes, viz., application and initial checks, regulation compliance checks, decision and administration, and enforcement, with clear dependencies between them. One sub-process cannot start until the previous one finished. So the dependencies between activities are primarily manifested between one sub-process and the initial few activities of its immediate successor. The event log contains 25 event classes (distinct activities) with each sub-process on average defined over 6 activities. Since the dependencies are mostly reflected in one sub-process and the initial few activities of the next sub-process, a window size of 10 is deemed appropriate. In fact, we have tried using other window sizes larger than 10 as well; however, we did not notice any difference in performance with respect to change detection and change localization. Since a smaller window size is computationally efficient, we reported the results on window size of 10.

The J-measure values of each activity pair define a vector of size 116, corresponding to the traces in the event log. The univariate Kolmogorov-Smirnov test (KS-test) is applied on each of these vectors using a population size of 10. Fig. 3 depicts the average significance probability of the KS-test on all activity pairs. We see four troughs formed at indices 42, 74, 84, and 103. These troughs signify a change in behavior in the traces preceding and succeeding them. Among the four troughs, the one at index 42 is particularly significant. Fig. 3 also depicts the start timestamps (04-10-2004, 27-10-2005, 13-02-2006, and 31-08-2006 respectively) of the cases corresponding to these troughs.

Based on the four change points, we split the log into five partitions, the first, $\mathcal{L}_1$, containing the traces from the beginning until the first change point (i.e., traces 1 to 42), the second, $\mathcal{L}_2$, containing the traces between the first and second change points (i.e., traces 43 to 74) and so on. Fig. 4 depicts the process model discovered using the Heuristic miner [12] on the event log $\mathcal{L}_1$. The process can be divided into four high-level sub-procedures as depicted in the figure and are listed below:

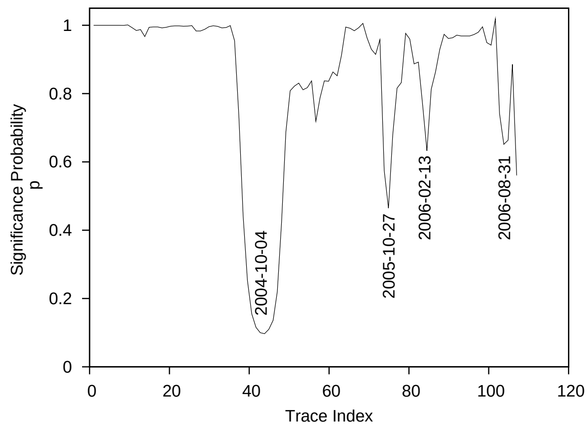


Fig. 3. Average significance probability (over all activity pairs) of KS-test on J-measure. The population size for the KS-test is 10. There are four troughs signifying a change in behavior.

- Upon submission of an application, the municipality acknowledges the receipt of documents and (optionally) tests for its completeness.
- The municipality then proceeds with a follow-up procedure that verifies whether the application and submitted documents are in compliance with the regulations.
- Based on the investigations, the municipality then makes a decision on the application and informs the applicant with the decision along with a fee letter.
- Finally, the municipality registers the advertisements placed and enforces them.

Fig. 5 depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_2$. The figure highlights regions that differ from the process model in Fig. 4. There are two changes in this model with respect to the previous one. The first change is related to the checking for completeness of the registered documents. In the initial process model (Fig. 4), this check was not mandatory (only 2 of the 43 applications were checked for completeness). The municipality changed this process by making the checks mandatory before proceeding. The second change is the introduction of a new activity End procedure: enforcement is next as highlighted in Fig. 5. The initial process model had only the activity End procedure, possibly choose enforcement where as the new model has both these activities.

Fig. 6(a) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_3$. This model contains only one type of enforcement activity End procedure: enforcement is next indicating that the municipality has phased out the activity End procedure, possibly choose enforcement. Fig. 6(b) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_4$. The change corresponds to the region marked in the figure involving the activi-

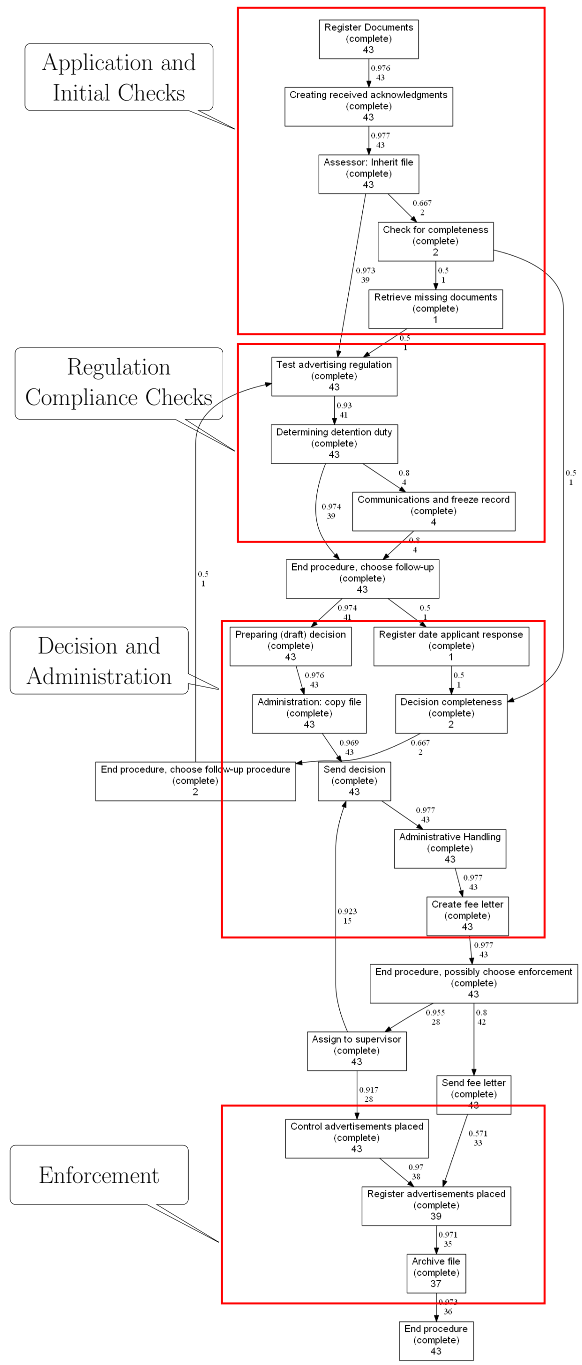


Fig. 4. Heuristic net of the permit process for advertisements discovered using the event log $\mathcal{L}_1$. The marked regions depict high-level sub-procedures in this process.

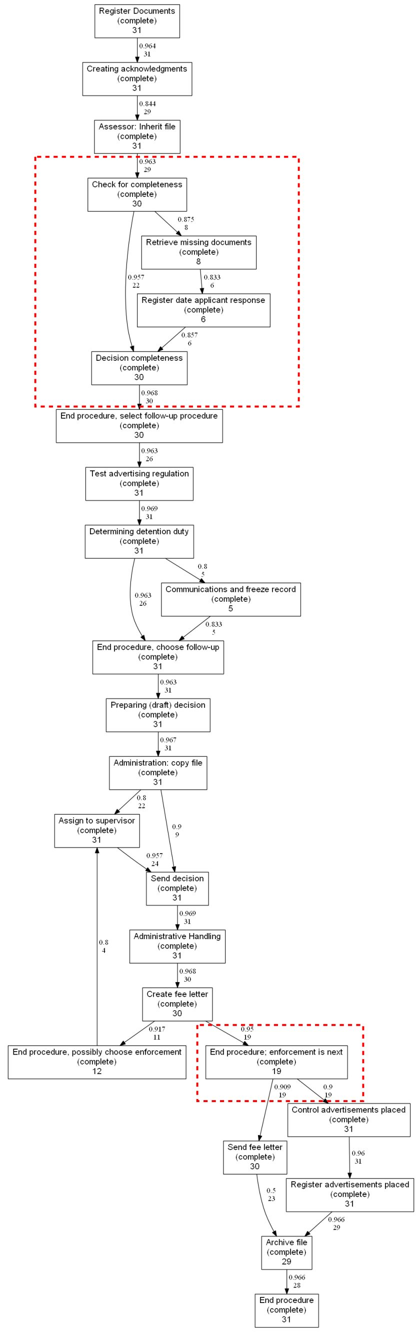


Fig. 5. Heuristic net of the permit process for advertisements discovered using the event log $\mathcal{L}_2$. The marked regions depict regions corresponding to the change in the process when compared to the model in Fig. 4. The municipality has now made the checks for completeness mandatory and introduced a new activity End procedure, enforcement is next.

ties Administration: copy file and Assign to supervisor. Unlike the previous model where these activities happen in a sequence, they can now be executed concurrently. Fig. 7 depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_5$. In all of the previous models, the activity Control advertisements placed is executed after the initiation of the enforcement procedure. However, in the model based on $\mathcal{L}_5$, this activity can execute concurrently with the administrative activities once a decision has been taken.

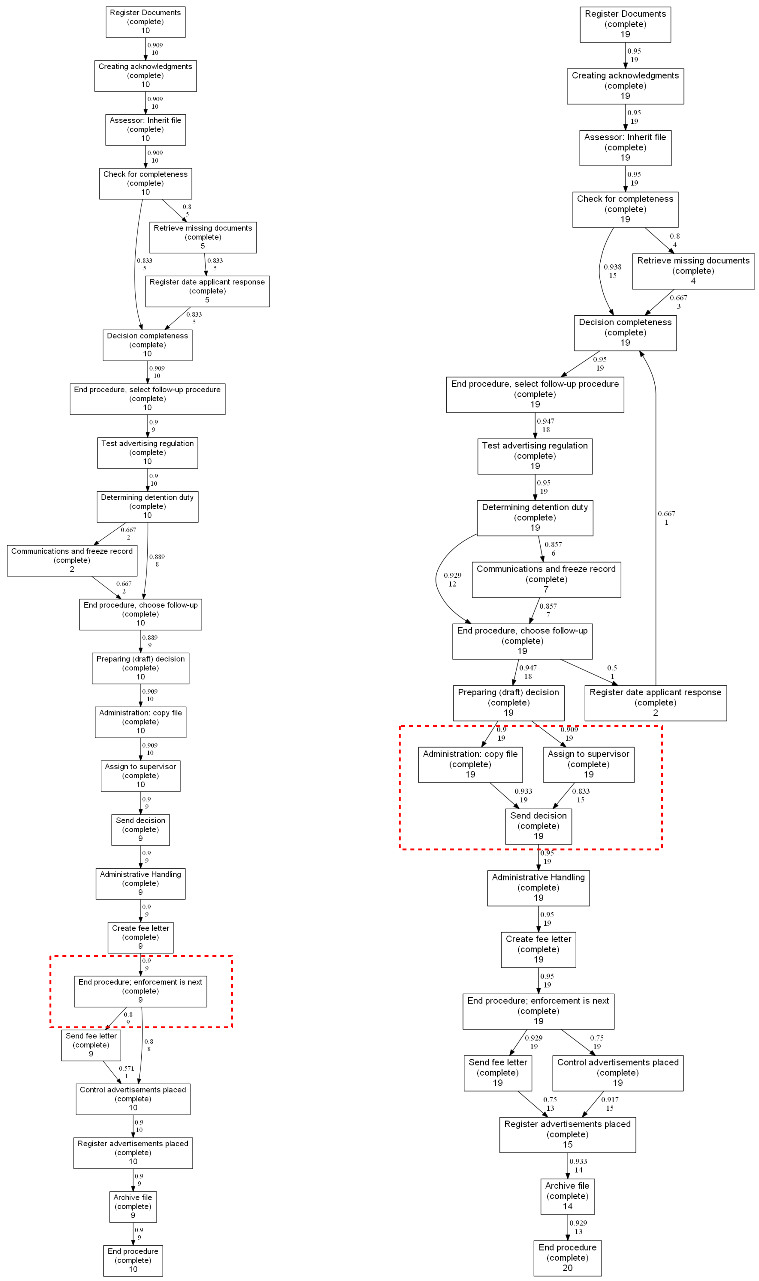
3.2 Permit Process for Temporary Rental of Vacant Dwellings

This process corresponds to obtaining permits for temporary rental of vacant dwellings. If a person wants to rent out unoccupied dwellings, a permit from the municipal authorities is required. A permit is sanctioned, usually valid for a period of two years, if they satisfy a number of conditions.

We considered an event log containing 35 cases and 315 events referring to 10 activities. The cases pertain to permit requests for temporary rental of vacant dwellings spanning over the period between 16-04-2009 and 05-01-2011. We considered the window count feature on the *follows* relation for all activity pairs using a window of size 4. Since the mean trace length is small (9), we considered a smaller window size for feature extraction. The window count feature of each activity pair defines a vector of size 35, corresponding to the traces in the event log. The univariate Kolmogorov-Smirnov test (KS-test) is applied on each of these vectors using a population size of 6 (since we have only 35 traces, we considered smaller populations). Fig. 8 depicts the average significance probability of the KS-test on all activity pairs. We see two troughs formed at indices 11 and 17. These troughs signify a change in behavior in the traces preceding and succeeding them. Fig. 8 also depicts the start timestamps (24-11-2009 and 12-02-2010 respectively) of the cases corresponding to these troughs.

Considering the two change points, we split the log into three partitions, the first, $\mathcal{L}_1$, containing the traces from the beginning until the first change point (i.e., traces 1 to 11), the second, $\mathcal{L}_2$, containing the traces between the first and second change points (i.e., traces 12 to 17), and the third, $\mathcal{L}_3$, containing the traces from the second change point until the end (i.e., traces 18 to 35). Fig. 9 depicts the process model discovered using the Heuristics miner [12] on the event log $\mathcal{L}_1$. The documents submitted by the applicant are first registered at the municipality (by an employee). The municipality notifies the applicant of the receipt of the documents and tests for its completeness. The municipality requests for the lease (rental agreement) of the vacant dwelling (if needed) and a decision is taken and communicated to the applicant. A fee letter is also prepared and sent. The preparation of the fee letter can happen before/after the decision is sent; this is captured in the parallel construct in Fig. 9. The sending of the fee letter can happen before/after the End procedure activity.

Fig. 10(a) depicts the process model discovered using the Heuristics miner on



(a) event log $\mathcal{L}_3$ (b) event log $\mathcal{L}_4$

Fig. 6. Heuristic nets of the permit process for advertisements discovered using the event logs $\mathcal{L}_3$ and $\mathcal{L}_4$. The activity End procedure, possibly choose enforcement has been phased out in (a) when compared to the model in Fig. 5. Unlike the model in (a), the activities Administration: copy file and Assign to supervisor can be executed in parallel in (b).

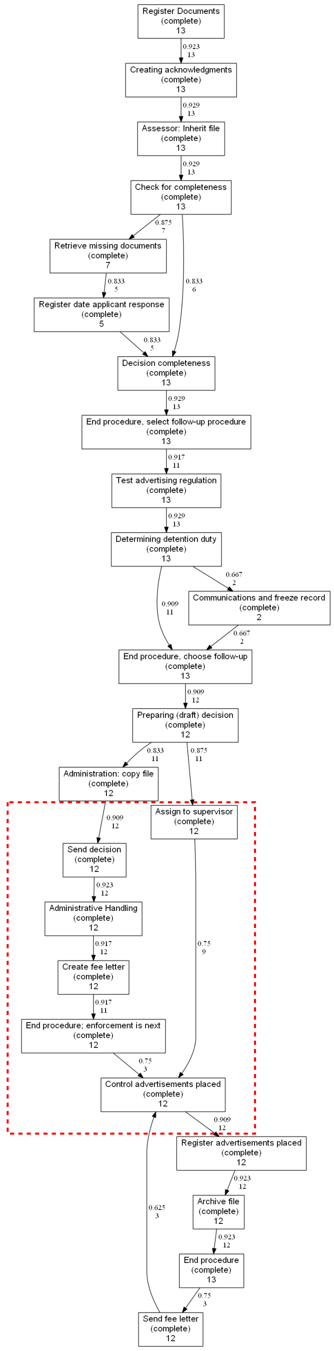


Fig. 7. Heuristic net of the permit process for advertisements discovered using the event log $\mathcal{L}_5$. Unlike the previous model in Fig. 6(b), the controlling of advertisements placed can now happen in parallel with the administrative activities once a decision has been taken.

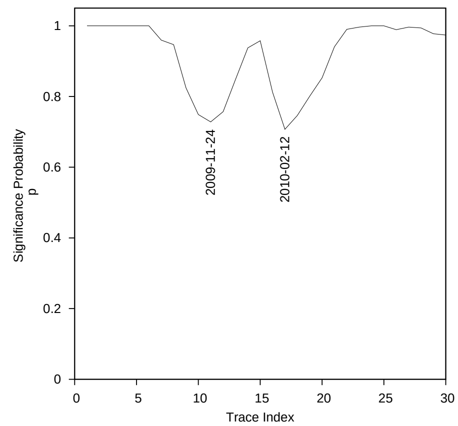


Fig. 8. Average significance probability (over all activity pairs) of KS-test on window count measure. The population size for the KS-test is 6. There are two troughs signifying a change in behavior.

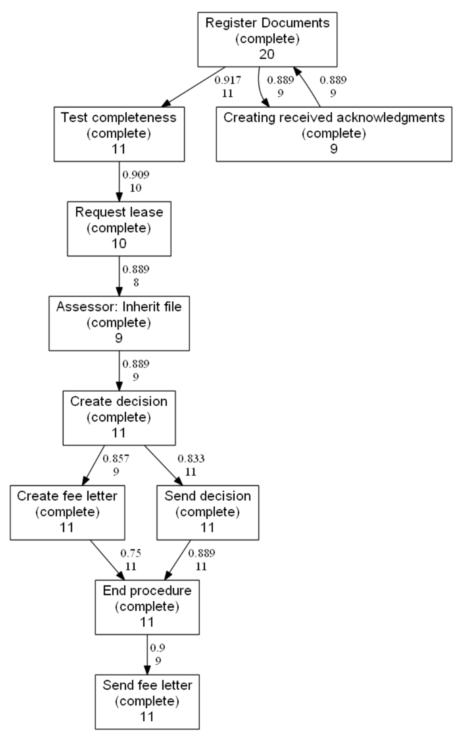
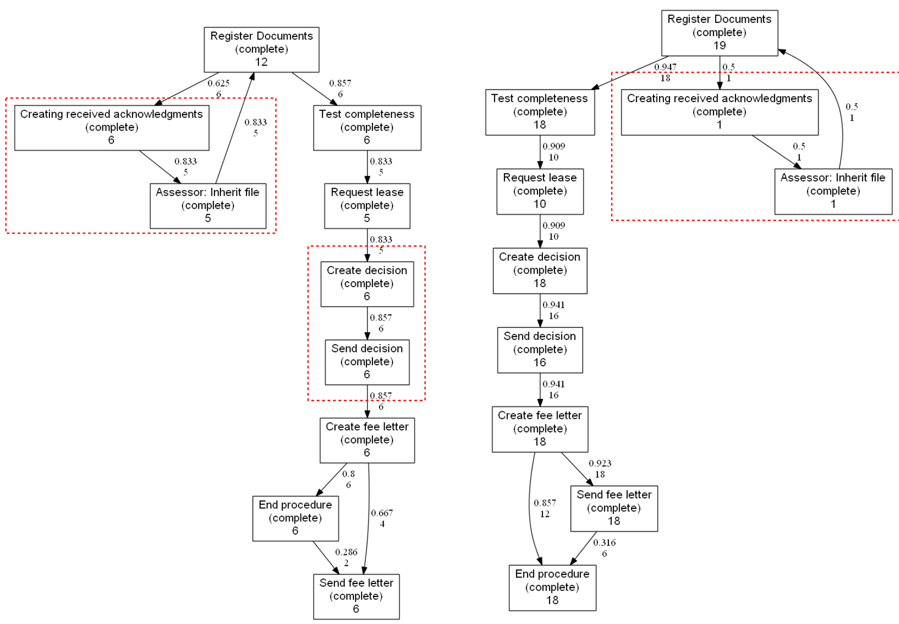


Fig. 9. Heuristic net of the permit process for temporary rental of vacant dwellings discovered using the event log $\mathcal{L}_1$.

the event log $\mathcal{L}_2$. There are two changes (marked with dashed rectangles) in this model when compared to the model in Fig. 9. Firstly, the activity Assessor: Inherit file related to Creating received acknowledgments has to be executed before Test completeness. Secondly, the creation of fee letter can happen only after the decision is sent. The process owners indeed acknowledged that their permit process has changed in November 2009, thus, validating our change point detection. Fig. 10(b) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_3$. The figure also depicts the region corresponding to the change when compared to Fig. 10(a). Unlike the process in Fig. 10(a), the municipality now does not send acknowledgements for each and every permit request. Out of the 18 permit requests only one of them was sent an acknowledgment (an applicant is allowed an option to indicate that a confirmation is not needed). Such measures are typically taken to reduce costs.

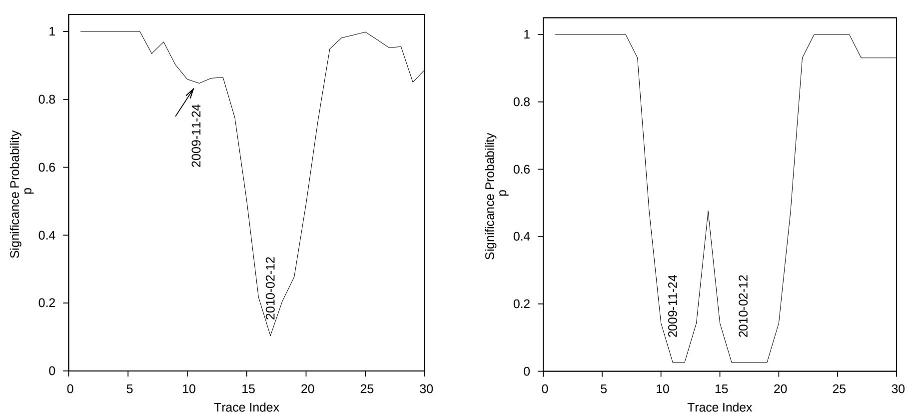


(a) event log $\mathcal{L}_2$

(b) event log $\mathcal{L}_3$

Fig. 10. Heuristic nets of the permit process for temporary rental of vacant dwellings discovered using the event logs $\mathcal{L}_2$ and $\mathcal{L}_3$. The dashed rectangles in (a) highlight the regions corresponding to the change in the process with respect to the process model in Fig. 9. The dashed rectangle in (b) highlights the region corresponding to the change in the process with respect to the model depicted in (a).

Fig. 11(a) depicts the average significance probability of the KS-test over all activity pairs for the same event log using the J-measure as the feature computed on a window of size 4. Unlike Fig. 8, we see only one trough. This can be attributed to the reliance of J-measure on the probability of activities. Since the activities Creating received acknowledgments and Assessor: Inherit file are rarely executed in the traces 18 to 35, the probability of activities is significantly different from that of the activities in the traces 1 to 17. This is captured in the trough at index 17. The probability of activities dominates the probability of the follows relation between activity pairs and hence we do not see a prominent change at index 11. Furthermore, since the significance probabilities are averaged over all activity pairs, the change at index 11 being confined to a few activities is obscured by the others. Nonetheless, we can see a minor dip at index 11 (as indicated by the arrow) in Fig. 11(a). Fig. 11(b) depicts the significance probability of the KS-test on the J-measure for the *follows* relation for the activity pair (Register Documents, Create Decision). We can now see two drifts signifying change points at indices 11 and 17. The change point at index 11 indicates that there is a change in the process in the region between Register Documents and Create Decision, which is indeed the case.



(a) (b)

Fig. 11. (a) Average significance probability (over all activity pairs) of KS-test on the J-measure. The population size for the KS-test is 6. There is one trough signifying a change in behavior. (b) Significance probability of the KS-test on the J-measure for the *follows* relation for the activity pair (Register Documents, Create Decision).

3.3 Permit Process for Driveway Construction

We analyzed for any concept drifts in the process pertaining to obtaining a permit to build or change a driveway from ones premises to the municipality's public road, referred to as a 'driveway permit' (inritvergunning) or as an 'egress permit' (uitwegvergunning). If a person (business) wants to build a driveway road to a provincial highway, permission from the provincial authority is needed. The municipality or provincial authority considers issues such as safe and efficient road use, protection of green spaces, and protection of the ambience of the surrounding areas before sanctioning a permit. The permit application must be accompanied with the necessary fees and documentation, including design plans, photos, and pertinent reports.

We considered an event log containing 315 cases and 3968 events referring to 21 activities. The cases pertain to permit requests for driveway construction spanning over the period between 03-01-2006 and 12-01-2011. We considered the J-measure feature on the *follows* relation for all activity pairs using a window of size 10. The J-measure values of each activity pair define a vector of size 315, corresponding to the traces in the event log. The univariate Kolmogorov-Smirnov test (KS-test) is applied on each of these vectors using a population size of 20. Fig. 12 depicts the average significance probability of the KS-test on all activity pairs. We see seven troughs formed at indices 21, 91, 125, 168, 213, 237, and 291 respectively. These troughs signify a change in behavior in the traces preceding and succeeding them. Based on these change points, we split the log into 8 partitions as depicted in the figure.

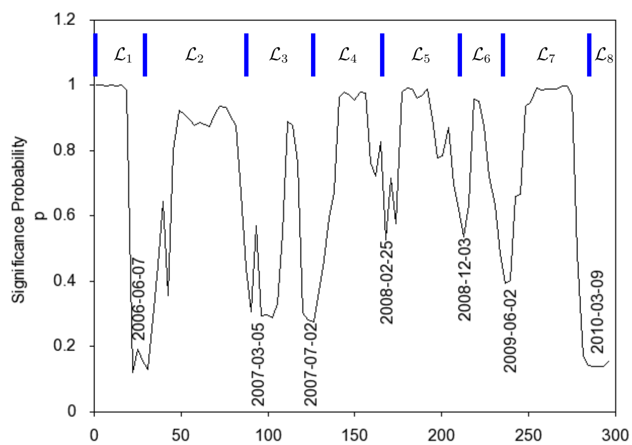


Fig. 12. Average significance probability (over all activity pairs) of KS-test on J-measure. The population size for the KS-test is 20. There are seven troughs signifying a change in behavior. These are used to partition the log into $\mathcal{L}_1$, $\mathcal{L}_2$, ..., $\mathcal{L}_8$.

Fig. 13 depicts the process model discovered using the Heuristics miner [12] on the event log $\mathcal{L}_1$. Upon submission of an application, the municipality acknowledges the receipt of documents and (optionally) tests for its completeness. A fee letter is also prepared and sent to the applicant. The municipality then proceeds with a follow-up procedure where an opinion from the department of planning

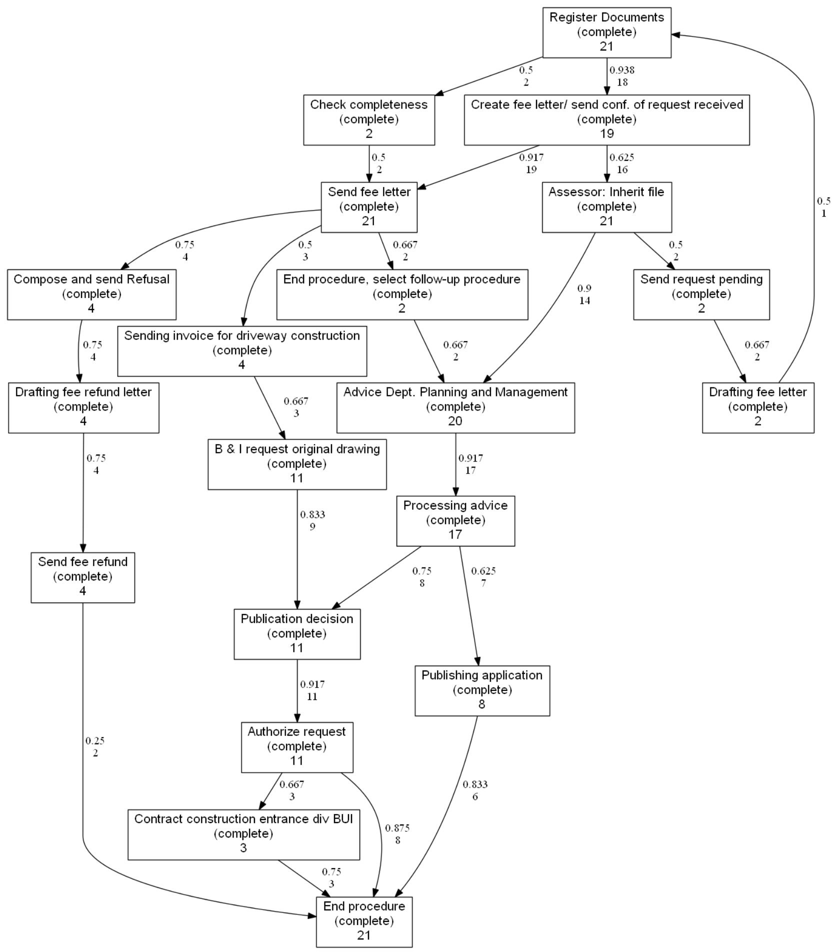


Fig. 13. Heuristic net of the permit process for driveway construction discovered using the event log $\mathcal{L}_1$.

and management is sought. If needed, the municipality may ask for the original drawings of the planned driveway. Based on the investigations, the municipality then makes a decision on the application and publishes them. The municipality may refuse the permission in which case a fee refund letter is sent to the applicant. For some requests that have been authorized, the municipality may send an invoice for driveway construction and contract the BUI division for construction. The BUI division may in turn confirm its decision to the municipality.

Fig. 14 depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_2$. There are two primary changes (marked by the dashed rectangles) in this process model when compared to the model in Fig. 13. Firstly, the checking for completeness of the registered documents is no longer optional. Also,

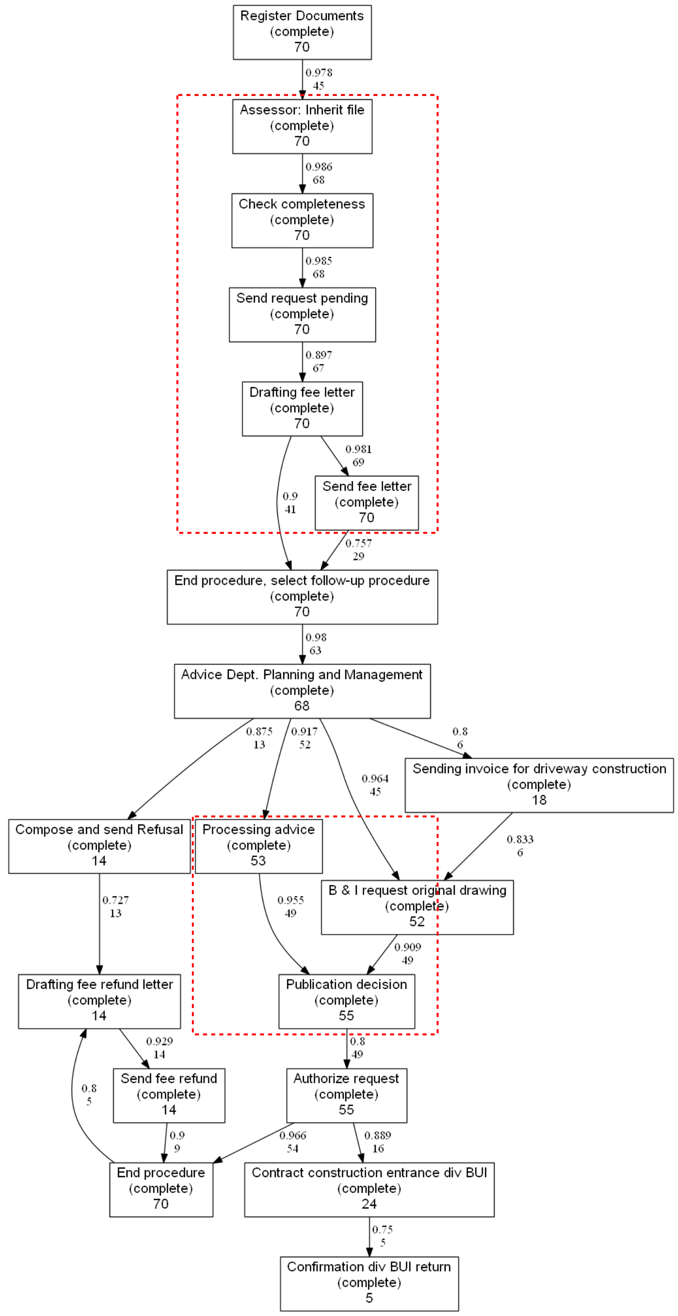


Fig. 14. Heuristic net of the permit process for driveway construction discovered using the event log $\mathcal{L}_2$. The dashed rectangles indicate regions where changes had taken place when compared to the previous model. When compared to the model in Fig. 13, the tests for completeness is now made mandatory and the confirmation of requests is no longer sent to applicants.

unlike the previous model where there are two activities related to creating of a fee letter, viz., Create fee letter/send conf. of request received and Drafting fee letter, we now have only one activity Drafting fee letter. In addition, the confirmation of requests is no longer sent to the applicants. The second change corresponds to the omission of the activity Publishing application. Fig. 15 depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_3$. There is one behavioral change in this model when compared to the one in Fig. 14. The municipality has now enforced a stricter need for confirmation from the BUI division. In Fig. 14, only 5 of the 24 contracts submitted to the BUI division were confirmed where as in Fig. 15, 13 of the 14 contracts were confirmed.

Fig. 16 depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_4$. There are two primary changes (as indicated by the dashed rectangles) in this model when compared to the previous model in Fig. 15. Firstly, a refund is recorded only to some selected cases in case of refusal of a permit. Secondly, the number of contracts given to the BUI division has drastically reduced. Fig. 17(a) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_5$. There are no further contracts to the BUI division in this process. Furthermore, the activities Advice Dept. Planning and Management and B & I request original drawing happen more often in parallel with the activities Send fee letter and End procedure, select follow-up procedure when compared to the previous model. Fig. 17(b) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_6$. In this model, the activities Drafting fee letter and Send request pending can be executed in parallel when compared to the previous models where they happen in a sequence. Furthermore, refunds have been stopped for all applicants in case of permit refusal.

Fig. 18(a) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_7$. This model differs from the model in Fig. 17(b) in that the activity Send fee letter has to happen before the End procedure, select follow-up procedure. Fig. 18(b) depicts the process model discovered using the Heuristics miner on the event log $\mathcal{L}_8$. The change in this model pertains to the activity Send fee letter. Unlike all of the previous models, the fee letter is sent only to some of the applicants (for just 6 of the 24 permit requests was a fee letter sent). Due to this, the probability of activities differ in the traces in $\mathcal{L}_8$ when compared to the traces in $\mathcal{L}_7$. The J-measure elegantly captures this and reflects this as a change in behavior.

4 Conclusions

In this paper, we analyzed event logs of three processes from a large Dutch municipality for the presence of concept drifts (i.e., process changes). The detection of such change points can help us put the results of process mining in a right perspective and enables an organization to take appropriate measures when a change in behavior is perceived. Using the framework proposed in [4] for dealing

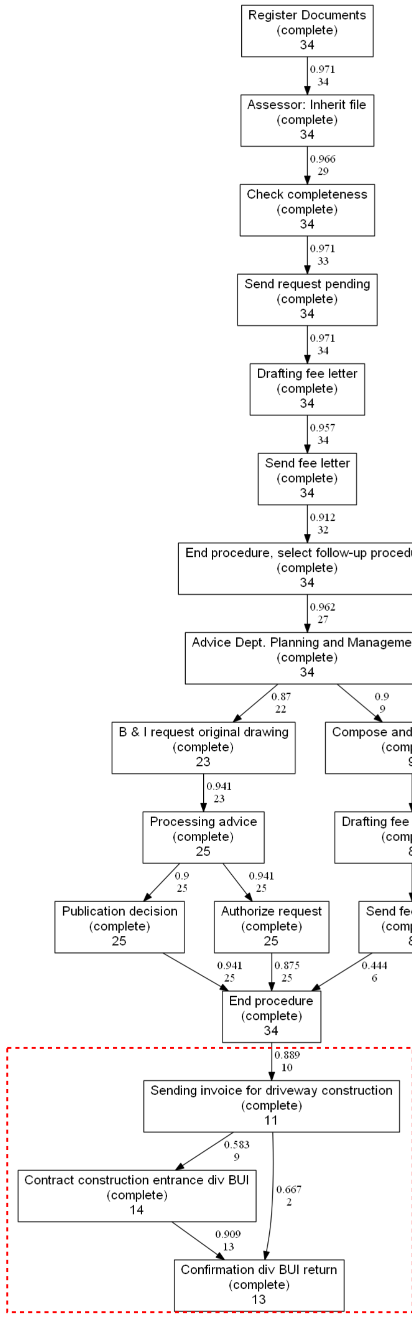


Fig. 15. Heuristic net of the permit process for driveway construction discovered using the event log $\mathcal{L}_3$. There is a stricter need for confirmation from the BUI division in this model when compared to the model in Fig. 14.

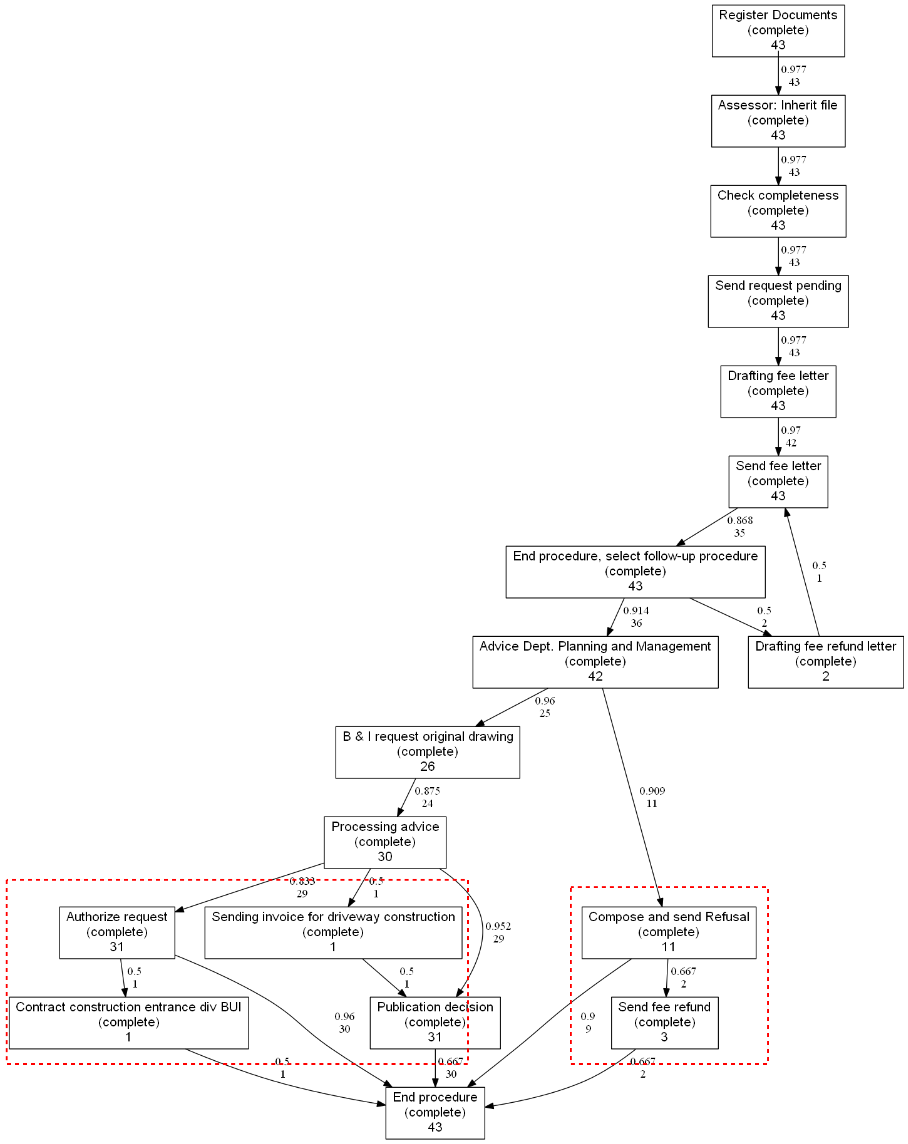
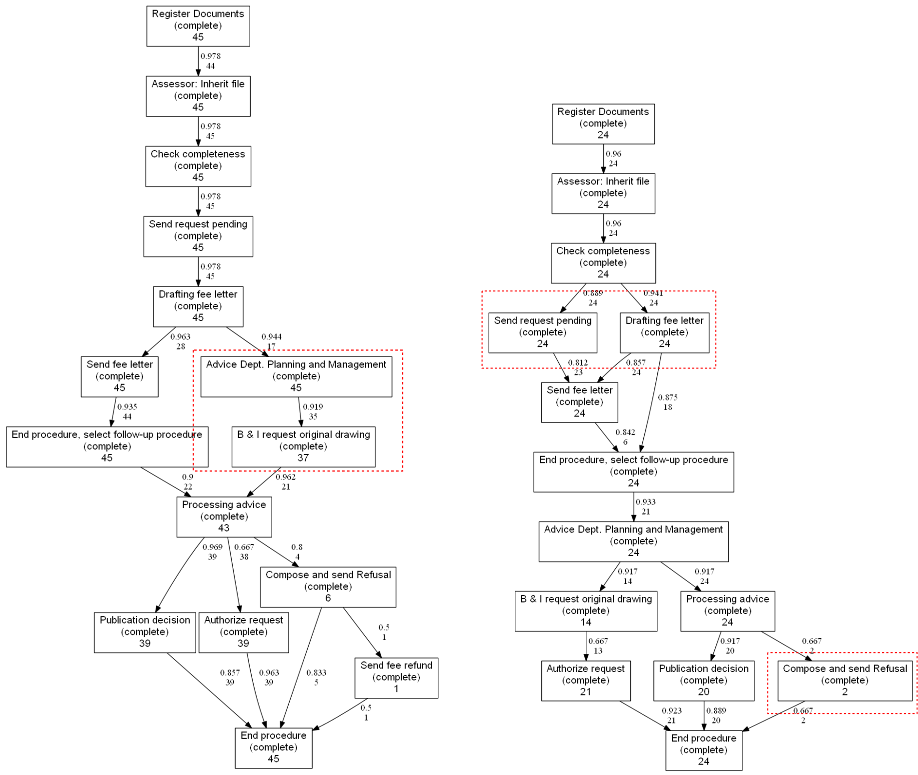


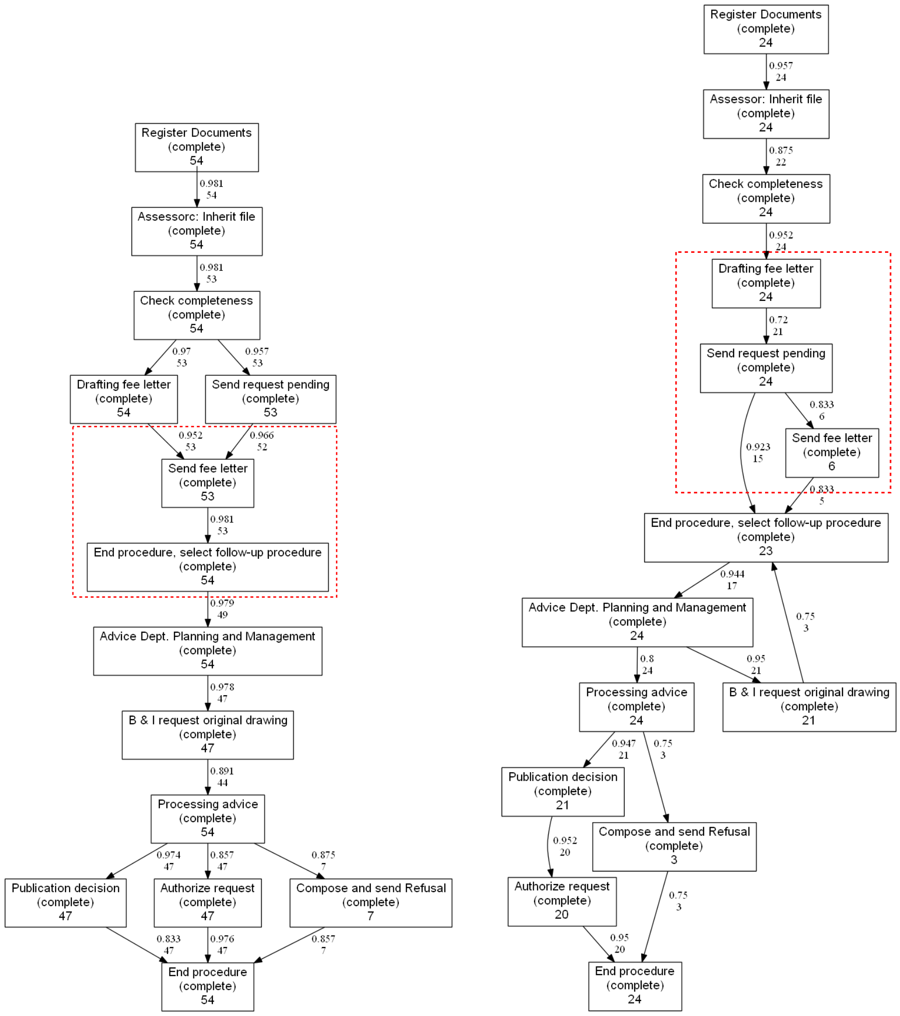
Fig. 16. Heuristic net of the permit process for driveway construction discovered using the event log $\mathcal{L}_4$. Refunds are done only for selected cases in case of permit refusal and the number of contracts given to the BUI division has reduced drastically when compared to the model in Fig. 15.



(a) event log $\mathcal{L}_5$ (b) event log $\mathcal{L}_6$

Fig. 17. Heuristic nets of the permit process for driveway construction discovered using the event logs $\mathcal{L}_5$ and $\mathcal{L}_6$. The marked regions signify the regions where changes are perceived when compared to the previous model. In the model in (a), the activities Advice Dept. Planning and Management and B & I request original drawing happen more often in parallel with Send fee letter and End procedure, select follow-up procedure, unlike the model in Fig. 16. Furthermore, the municipality no longer assigns contracts to the BUI division. The activities Drafting fee letter and Send request pending can be executed in parallel in (b) when compared to the model in (a).

with concept drifts in process mining, we are able to detect changes in real-life event logs even with a small number of cases.



(a) event log $\mathcal{L}_7$ (b) event log $\mathcal{L}_8$

Fig. 18. Heuristic nets of the permit process for driveway construction discovered using the event logs $\mathcal{L}_7$ and $\mathcal{L}_8$. The sending of fee letter has to happen before the activity End procedure, select follow-up procedure in (a) while the activity is optional in (b).

References

1. van Infrastructuur en Milieu, M.: All-in-one permit for physical aspects: (omgevingsvergunning) in a nutshell (2010)
2. Sarbanes, P., G. Oxley et. al.: Sarbanes-Oxley Act of 2002 (2002)
3. van der Aalst, W.M.P., Rosemann, M., Dumas, M.: Deadline-based Escalation in Process-Aware Information Systems. Decision Support Systems **43**(2) (2011) 492–511
4. Bose, R.P.J.C., van der Aalst, W.M.P., Žliobaitė, I., Pechenizkiy, M.: Handling Concept Drift in Process Mining. In Mouratidis, H., Rolland, C., eds.: International Conference on Advanced Information Systems Engineering (CAiSE 2011). Volume 6741 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2011) 391–405
5. Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC (2004)
6. Bose, R.P.J.C.: Process Mining in the Large: Preprocessing, Discovery, and Diagnostics. PhD thesis, Eindhoven University of Technology (2012)
7. CoSeLog: Configurable Services for Local Governments Project Home Page. `www.win.tue.nl/coselog`.
8. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards Cross-Organizational Process Mining in Collections of Process Models and thier Executions. In Daniel, F., Barkaoui, K., Dustdar, S., eds.: Business Process Management Workshops. Volume 100 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2012) 2–13
9. van der Aalst, W.M.P.: Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining. In Meersman, R., Dillon, T., Herrero, P., eds.: On the Move to Meaningful Internet Systems: OTM 2010. Volume 6426 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2010) 8–25
10. van der Aalst, W.M.P.: Intra– and Inter–Organizational Process Mining: Discovering Processes Within and Between Organizations. In Johannesson, P., Krogstie, J., Opdahl, A.L., eds.: The Practice of Enterprise Modeling (PoEM). Volume 92 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2011) 1–11
11. Vogelaar, J.J.C.L., Verbeek, H.M.W., Luka, B., van der Aalst, W.M.P.: Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study. In Daniel, F., Barkaoui, K., Dustdar, S., eds.: Business Process Management Workshops. Volume 100 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2012) 50–61
12. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models From Event-based Data Using Little Thumb. Integrated Computer-Aided Engineering **10**(2) (2003) 151–162