# ImplementationofConceptDriftsin DataMiningProcess

**Ganapam Venkata Suresh Reddy[1] Pasupula Venkata Nagendra Kumar[2]**
[1]System Trainee [2]Sr.Tech Associate
[1]Polaris Financial Technology Ltd.[2]Bank of America

**Abstract**—*Eventhough most business processes change over time, contemporary process mining techniques tendtoanalyze theseprocessesasiftheyareinasteadystate.Processesmaychange suddenly or gradually. The drift may beperiodic (e.g.,because of seasonal influences)or one-of-a-kind(e.g. theeffects of new legislation). For the process management,itiscrucial to discover and understand such concept drifts in processes. This paper presents a generic frameworkandspecific techniques todetect when a process changes andtolocalize the parts of the process that have changed. Different features are proposedto characterize relationshipsamong activities. These features areused to discover differences between successivepopulations.The approach has been implemented as a plug-in of the ProM process mining framework and has been evaluated using both simulated event data exhibiting controlled concept driftsandreal-lifeeventdatafromaDutchmunicipality.*

**Keywords**—Concept drift, litheness, hypothesis tests,processchanges,processmining

## I. INTRODUCTION

USINESS processes are nothing more than logically related tasks that use the resources of an organization to achieve a defined business outcome. Business processes can be viewed from a number of perspectives, including the control flow, data, and the resource perspectives. In today's dynamic marketplace, it is increasingly necessary for enterprises to streamline their processes so as to reduce cost and to improve performance. In addition, today's customers expect organiza- tions 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], and so on, 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 of work in the pipeline. As another example, in a disaster, hospitals, and banks change their operating procedures. 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. Therefore, flexibility and change have been studied in-depth in the context of business process management (BPM). For example, process-aware information systems (PAISs) [4] have been extended to be able to flexibly adapt to changes in the process. State-of-the-art workflow management (WFM) and BPM systems [5] provide such flexibility, e.g., we can easily release a new version of a process. In addition, in processes not driven by WFM/BPM systems (such as the usage of medical systems) there is even more flexibility as processes are controlled by people rather than information systems.

Many of today's information systems are recording an abundance of event logs. Process mining is a relatively young research discipline aimed at discovering, monitoring, and improving real processes by extracting knowledge from event logs [6] (Section II-A for a brief introduction). Although flexibility and change have been studied in-depth in the context of WFM and BPM systems, contemporary process mining techniques assume the processes to be in a steady state. For example, when discovering a process model from event logs, it is assumed that the process at the beginning of the recorded period is the same as the process at the end of the recorded period. Using ProM,1 we have analyzed processes in more than 100 organizations. These practical experiences show that it is very unrealistic to assume that the process being studied is in a steady state. As mentioned earlier, processes may change to adapt to changing circumstances. 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 obtain an accurate insight on process executions at any instant of time. When dealing with concept drifts in process mining, the following three main challenges emerge.

*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), we 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, and so on) and the identification of the exact change itself. For instance, in the example of a seasonal process, the change could bethatmoreresources aredeployedorthatspecialoffers areprovidedduringholidayseasons.

*3) Changeprocessdiscovery:*
Havingidentified,localized, andcharacterizedthechanges,itisnecessarytoputallof theseinperspective.Thereisaneedfortechniques/tools thatexploitandrelatethesediscoveries. Unravelingthe evolution ofaprocessshouldresultinthediscovery of thechangeprocessdescribingthesecond-order dynam- ics.Forinstance,intheexample ofaseasonal process, wecouldidentifythattheprocessrecurseveryseason. In addition,we can show an animationon how the

processevolved overaperiodwithannotationsshowing severalperspectivessuch as the performancemetrics (servicelevels,throughputtime,andsoon)ofaprocess atdifferentinstancesoftime.

Wecandifferentiatebetweentwobroadclassesofdealing withconceptdriftswhenanalyzingeventlogs(Fig.1).

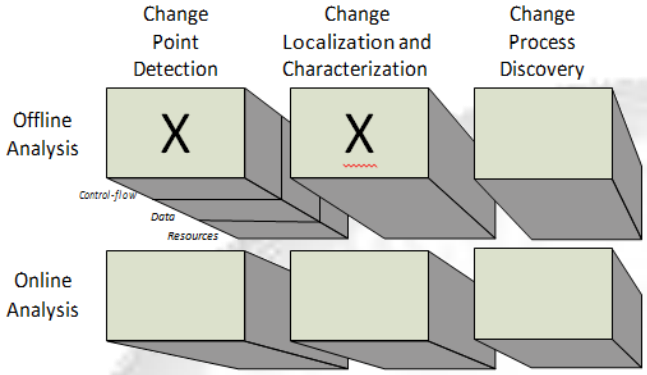



Fig. 1:
Differentdimensionsofconceptdriftanalysisinprocessmining.

1) Offline analysis: This refers to the scenario where the presence of changes or the occurrence of drifts need not be uncovered in a real time. This is appropriate in cases where the detection of changes is mostly used in postmortem analysis, the results of which can be considered when designing/improving processes for later deployment. For example, offline concept drift analysis can be used to better deal with seasonal effects (hiring less staff in summer or skipping checks in the weeks before Christmas).

2) Online analysis: This refers to the scenario where changes need to be discovered in near real time. This is appropriate in cases where an organization would be more interested in knowing a change in the behavior of their customers or a change in demand as and when it is happening. Such real-time triggers (alarms) will enable organizations to take quick remedial actions and avoid any repercussions.

In this paper, we focus on two of the challenges: 1) change (point) detection and change localization and 2) characterization in an offline setting (Fig. 1). We define different features and propose a framework for dealing with these two problems from a control-flow perspective. Initially, we show the promise of the techniques proposed in this paper on a synthetic log and later evaluate them on a real-life case study from a large Dutch municipality.

The rest of this paper is organized as follows. Section II pro- vides background on process mining and concept drifts in data mining. Related work is presented in Section III. Section IV describes the various aspects and nature of change, whereas Section V presents the basic idea for change detection inevent logs. Section VI introduces various features that capture the characteristics of event logs. Section VII illustrates the significance of statistical hypothesis tests for detecting drifts. Section VIII presents the framework for dealing with concept drifts in process mining, whereas Section IX presents the realization of the proposed approaches in the ProM framework. Section X describes the effectiveness of the features and thetechniques proposed in this paper on a synthetic log as well as a real-life case study. Finally, this paper is summarized with a conclusion and an outlook on some of the open research questions in Section XI.

## II. BACKGROUND

In this section, we discuss the basic concepts in process mining and concept drifts in data mining/machine learning.

### *A. ProcessMining*

Process mining serves a bridge between data mining and business process modeling [6]. Business processes leave trails in a variety of data sources (e.g., audit trails, databases, and transaction logs). Process mining aims at discovering, moni- toring, and improving real processes by extracting knowledge from event logs recorded by a variety of systems (ranging from sensor networks to enterprise information systems). The start- ing point for process mining is an event log, which is a collec- tion of events. We assume that events can be related to process instances (often called cases) and are described by some activ- ity name. The events within a process instance are ordered. Therefore, a process instance is often represented as a trace over a set of activities. In addition, events can have attributes such as timestamps, associated resources (e.g., the person exe- cuting the activity), transactional information (e.g., start, com- plete, suspend, and so on), and data attributes (e.g., amount or type of customer). For a more formal definition of event logs used in process mining, the reader is referred to [6]. Fig. 2 shows a fragment of an example log. Event logs like in Fig. 2 are completely standard in the process mining community and event log formats such as MXML [7] and XES [8] are used.

The topics in process mining can be broadly classified into three categories: 1) discovery; 2) conformance; and

3) enhancement [6]. Process discovery deals with the discovery of models from event logs. These models may describe control flow, organizational aspects, time aspects, and so on. For example, there are dozens of techniques that automatically construct process models (e.g., Petri nets or BPMN mod- els) from event logs [6]. Fig. 2 shows the basic idea of process discovery. An event log containing detailed infor- mation about events is transformed into a multiset of traces

L = [abcdjkln, aefjkmn, abgchdjkln, ... ].

Process discovery techniques are able to discover process models such as the Petri net shown in Fig. 2. Conformance deals with comparing an a priori process model with the observed behavior as recorded in the log and aims at detecting inconsistencies/deviations between a process model and its corresponding execution log. In other words, it checks for any violation between *whatwasexpected tohappen* and*whatactually has happened*.Enhancementdealswithextendingorimproving an existingmodelbasedon informationabouttheprocess executioninaneventlog.Forexample,annotating aprocess modelwithperformancedatatoshowbottlenecks,throughput times,andsoon.

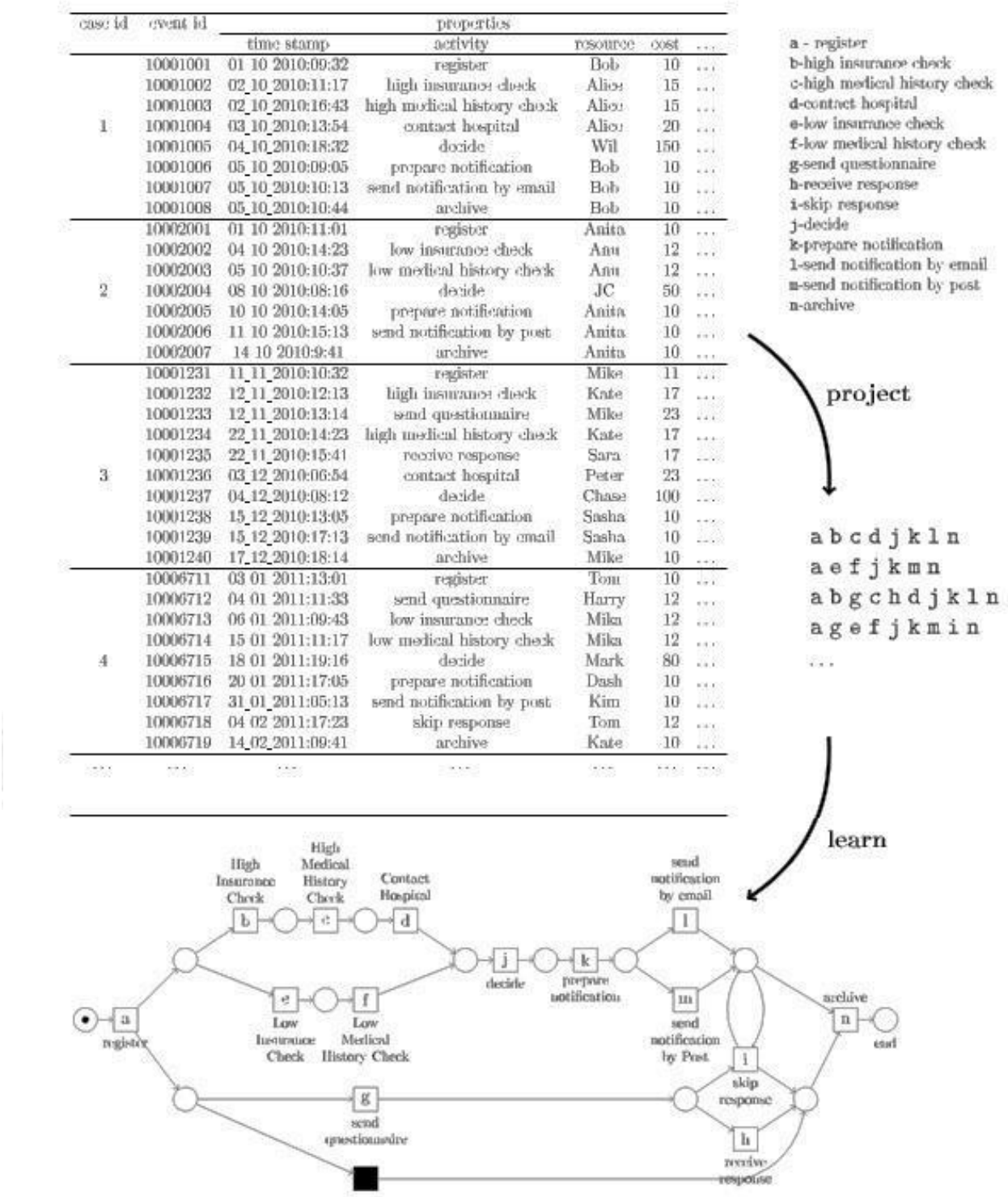



Fig. 2: Process discovery aims to learn a process model (in this case a Petri net) from event logs. An event log consists of events related to cases and referring to activities. To discover control flow, traces are projected onto activity names.

Beingarelativelyyoungresearchdiscipline,severalprocess miningchallengesremainto beaddressed.Theprocessmining manifesto [9]lists11challenges. Thefourthchallenge is dealingwithconcept driftand,thusfar,alittleworkhasbeen doneonthishighlyrelevanttopic[10],[11].

*B. ConceptDrift*

Conceptdrift[12]in machinelearninganddatamining referstosituationswhentherelationbetweentheinputdata and the target variable, which the model is trying to predict, changes over time in unforeseen ways. Therefore, the accuracy of the predictions may degrade over time. To prevent that, predictive models need to be able to adapt online, i.e., to update themselves regularly with new data.

The setting is typically looped over an infinite data stream as follows:

1) receive new data;
2) make a prediction;
3) receive feedback (the true target value); and

4) update the predictive model.

While operating under such circumstances, predictive models are required:

1) to react to concept drift (and adapt if needed) as soon as possible;
2) to distinguish drifts from once-off noise and adapt to changes, but be robust to noise; and
3) to operate in less than data arrival time and use limited memory for storage. In this setting, many adaptive algorithms have been developed (e.g., overviews [13], [14]).

Concept drift is a relatively young research topic that has gained popularity in data mining and machine learning communities in the last 10 years. Concept drift research primarily has been focusing on two directions: 1) how to detect drifts (changes) online (e.g., [15]–[20]) and 2) how to keep predictive models up to date (e.g., [21]–[23]). Concept drift has been shown to be important in many applications (e.g., [24]–[26]). The basis for drift detection could be a raw data stream, a stream of prediction errors, and, more rarely, a stream of predictions or a stream of updated model parameters. Two types of concept drift detection approaches have been used: monitoring evolution of a stream [15], [17] or comparing data distributions in two time windows [16], [18]. The cumulative sum (CUSUM) approach [27] is a representative sequential analysis technique for change detection, different extensions to which have been proposed. One notable example is computational intelligence-based CUSUM or CI-CUSUM [19] that aims to detect a non-stationarity condition by monitoring a multidimensional vector, i.e., multiple features. Adaptive windowing [16] is a representative approach for online change detection using an adaptive size sliding detection window. In this paper, we consider offline change detection and its localization and therefore focus on studying what features to monitor and how to identify when these characteristics change.

## III. Characterization of Changes in Business Processes

In this section, we discuss the various aspects of process change. Initially, we describe change perspectives (control flow, data, and resource). Then, the different types of drift (sudden, gradual, recurring, periodic, and incremental) are discussed.

### *A. Perspectives of Change*

There are three important perspectives in the context of business processes: 1) control flow; 2) data; and 3) resource. One or more of these perspectives may change over time.

*1) Control flow/behavioral perspective:*
This class of changes deals with the behavioral and structural changes in a process model. Just like the design patterns in software engineering, there exist change patterns cap- turing the common control-flow changes [29]. Control- flow changes can be classified into operations such as insertion, deletion, substitution, and reordering of process fragments. For example, an organization which used to collect a fee after processing and acceptance of an application can now change their process to enforce payment of that fee before processing an application. Here, the reordering change pattern had been applied on the payment and the application processing process fragments. As another example, with the addition of new product offerings, a choice construct is inserted into the product development process of an organization. In the context of PAISs, various control-flow change patterns have been proposed in [28], [29]. Most of these control-flow change patterns are applicable to traditional information/workflow systems as well.

Sometimes, the control-flow structure of a process model can remain intact but the behavioral aspects of a model change. For example, consider an insurance agency that classifies claims as high or low depending on the amount claimed. An insurance claim of e1000 which would have been classified as high last year is categorized as a low insurance claim this year because of the organization's decision to increase the claim limit. The structure of the process remains intact but the routing of cases changes.

*2) Dataperspective:*
Thisclass of changesreferto the changesintheproductionandconsumption ofdataand theeffectofdataontheroutingofcases.Forexample, itmaynolonger berequired tohaveaparticular documentwhenapprovingaclaim.

*3) Resource perspective:*
This class deals with the changes in resources, their roles, and organizational structure, and their influence on the execution of a process. For example, there could have been a change pertaining to who executes an activity. Roles may change and people may change roles. As another example, certain execution paths in a process could be enabled (disabled) upon the availability (nonavailability) of resources. Furthermore, resources tend to work in a particular manner and such working patterns may change over time, e.g., a resource can have a tendency of executing a set of parallel activities in a specific sequential order. Such working patterns could be more prominent when only few resources are available; the addition of new resources can remove this bias.

### *B. NatureofDrifts*

With the duration for which a change is active, we can classify changes into momentary and permanent. Momentary changes are short lived and affect only a very few cases, whereas permanent changes are persistent and stay for a while [31]. In this paper, we focus on permanent changes as momentary changes often cannot be discovered because of insufficient data.2 Momentary changes correspond to the notion of outliers/noise in data mining. Changes are perceived to induce a drift in the concept (process behavior). As shown in Fig. 3, we identify four classes of drifts.

*1) Suddendrift:*
This corresponds to a substitution of an existing process M1 with a new process M2 , as shown in Fig. 3(a). M1 ceases to exist from the moment of substitution. In other words, all cases (process instances) from the instant of substitution emanate from M2 . This class of drifts is typically seen in scenarios such as emergencies, crisis situations, and change of law. As an example, a new regulation by the finance ministry of India mandates all

banks to procure and report the customer's personal account number in their transactions.

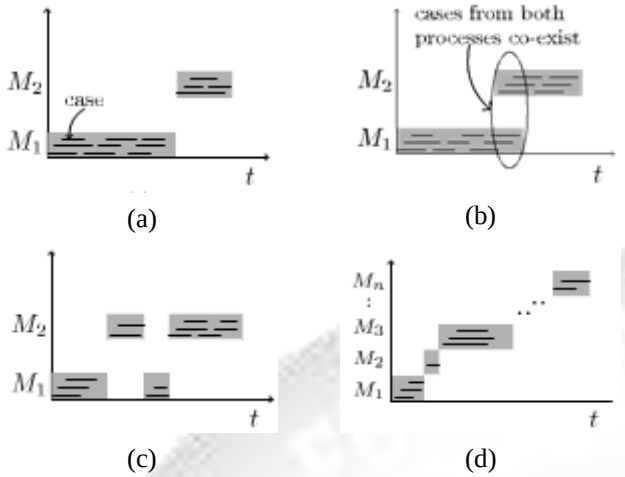



Fig. 3: Differenttypesofdrifts.*x*-axis:time.*y*-axis:processvariants.Shadedrectangles:processinstances.(a)Suddendrift.(b)Gradualdrift.(c)Recurring drift.(d)Incrementaldrift.

*2) Gradual drift:*

This refers to the scenario, as shown in Fig. 3(b) where a current process M1 is replaced with a new process M2 . Unlike the sudden drift, here both processes coexist for some time with M1 discontinued gradually. For example, a supply chain organization might introduce a new delivery process. This process is, however, applicable only for orders taken henceforth. All previous orders still have to follow the former delivery process.

*3) Recurringdrift:*

This corresponds to the scenario where a set of processes reappear after some time (substituted back and forth), as shown in Fig. 3(c). It is quite natural to observe such a phenomenon with processes having a seasonal influence. For example, a travel agency might deploy a different process to attract customers during Christmas period. The recurrence of processes may be periodic or nonperiodic. An example of a nonperiodic recurrence is the deployment of a process subjected to market conditions. The point of deployment and the duration of deployment are both dependent on external factors (here, the market conditions). Periodic drifts may be caused by seasonal effects, e.g., during the summer holidays there tends to be less demand and fewer resources thus influencing the process.

*4) Incrementaldrift:*

Thisreferstothescenariowherea substitution ofprocess$M_1$with$M_N$ isdoneviasmaller incrementalchanges,as shownin Fig.3(d).Thisclassof driftsismorepronouncedinorganizations adoptingan agileBPMmethodology andinprocessesundergoing sequencesofqualityimprovements (mosttotalquality management)initiatives are examples of incremental change[39]).

Recurring and incremental drifts in Fig. 3 are shown as discrete sudden changes. These two types of concept drift, however, can also be gradual. Similar categorization of drifts have been proposed in [40] in the context of machine learning. Drifts in [40] are further classified based on the severity of change into severe (and intersected). The categories of severity, as defined in [40], are too coarse to be applied to business process changes. Nonetheless, the degree of severity in process changes and their impact on dealing with concept drifts is an interesting topic for further research. In the rest, we propose approaches to detect potential control-flow changes in a process manifested as sudden/gradual drifts over a period. Detecting drifts in the other perspectives are beyond the scope of this paper. In addition, as already shown in Fig. 1, we focus on offline concept drift analysis (although our techniques can easily be adapted to the online setting). In practice, a mixture of any or all of the drifts may happen.

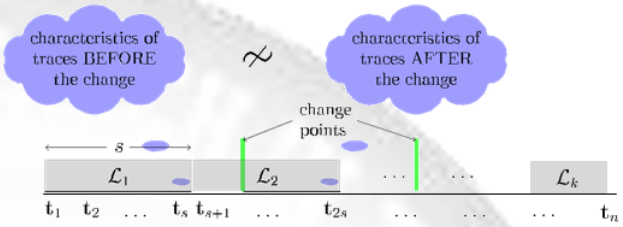



Fig. 4: Eventlogvisualizedas atimeseriesoftracesalongwithchange points.Thebasicpremise ofchange(point)detectionisthatcharacteristic differencesexistinthetracesbeforeandafterthechange

## IV. Basic Idea of Drift Detection in Event Logs

In this section, we present the basic idea for the detection of changes by analyzing event logs. Initially, we introduce the notations used in this paper.

1) $A$ isthesetofactivities.$A^+$ isthesetofallnonempty finitesequencesofactivitiesfrom$A$.

2) Aprocessinstance(i.e.,case)isdescribedasatrace over$A$,i.e.,afinitesequenceofactivities.Examples of tracesareabcdandabbbad.

3) Let$\mathbf{t}=\mathbf{t}(1)\mathbf{t}(2)\mathbf{t}(3)\ldots\mathbf{t}(n)\in A^+$ beatraceover$A$. $|\mathbf{t}|=n$isthelengthofthetrace$\mathbf{t}$.$\mathbf{t}(k)$isthe$k^{th}$activity inthetraceand$\mathbf{t}(i,j)$isthecontinuoussubsequenceof $\mathbf{t}$thatstartsatposition$i$ andendsatposition $j$.$\mathbf{t}^i=(i,|\mathbf{t}|)$representsthesuffixof$\mathbf{t}$thatbeginsat position $i$.

4) Aneventlog,$L$, correspondstoamultiset(orbag)of tracesfrom$A^+$.Forexample,$L$=[abcd, abcd,abbbad] isalogconsistingofthreecases.Two cases followtrace abcdandonecasefollowstraceabbbad.

5) N,$N_0$,and$R^+$ arethesetofallnaturalnumbers,the setofallnaturalnumbersincluding zero,andthesetof allpositiverealnumbersincludingzero,respectively.

Wecanconsideraneventlog$L$ asatimeseriesoftraces (traces ordered based on the timestamp of the first event). Fig. 4 shows such a perspectiveon an event log alongwithchangepointsinthesuddendriftscenario.

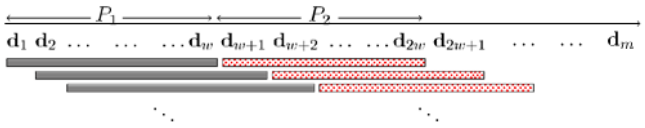



Fig.7: Basicideaofdetectingdriftsusinghypothesistests.Thedatasetof featurevaluesisconsideredasatimeseriesforhypothesistests.$P_1$ &$P_2$ aretwopopulationsofsize$w$.

## V. HYPOTHESIS TESTS FOR DRIFT DETECTION

An event log can be transformed into a data stream/sequence $D$ by choosing one of the feature sets defined in the previous section. The dataset $D$ of feature values can be considered as a time series of $m$ values, as shown in Fig. 7. Each $\mathbf{d}_i \in D$ corresponds to the feature value(s) for a trace (or sublog) and can be a scalar or a vector (depending on the choice of feature).[5] Comparing with Fig. 4, $m = n$ or $m = k$ depending on whether the feature values are computed for each trace or for each sublog, respectively. As mentioned earlier, we expect a characteristic difference in the manifestation of feature values in the traces (sublogs) before and after the change points with the difference being more pronounced at the boundaries. To detect this, we can consider a series of successive populations of values (of size $w$) and investigate if there is a significant difference between two subsequent populations. The premise is that differences are expected to be perceived at change points provided appropriate characteristics of the change are captured as features. A moving window of size $w$ is used to generate the populations. Fig. 7 shows a scenario where two populations $P_1 = \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_w$ and $P_2 = \mathbf{d}_{w+1}, \mathbf{d}_{w+2}, \ldots, \mathbf{d}_{2w}$ of size $w$ are considered. In the next iteration, the populations correspond to $P_1 = \mathbf{d}_2, \mathbf{d}_3, \ldots, \mathbf{d}_{w+1}$ and $P_2 = \mathbf{d}_{w+2}, \mathbf{d}_{w+3}, \ldots, \mathbf{d}_{2w+1}$. Given a dataset of $m$ values, the number of population pairs (iterations) will be $m - 2w + 1$.

We propose the use of statistical hypothesis testing to discover these change points. Hypothesis testing is a procedure in which a hypothesis is evaluated on a sample data. One of the important uses of hypothesis testing is to evaluate and compare groups of data. Numerous varieties of hypothesis tests exist [43]. The choice of a particular test is largely dependent on the nature of the data and the objectives of an experiment. For example, hypothesis tests can be classified into parametric and nonparametric tests. Parametric tests assume that the data have a particular distribution, e.g., normal, whereas the nonparametric tests do not make any assumption with regards to the data distribution. Because we do not know the *apriori* distribution of the feature values in an event log, we consider only nonparametric tests. Another perspective of classification is based on the number of samples (populations) on which the hypothesis is defined. We can classify the hypothesis tests into 1) one-sample; 2) two-sample; and 3) multisample tests. Because we need to analyze two populations for detecting drifts, we are interested in two-sample hypothesis tests. Another classification of hypothesis tests is concerned with the dimensionality of each data element in a sample. Tests dealing with scalar data elements are called univariate tests while those dealing with vector data elements are called multivariate tests. If only a particular activity or activity pair is considered, then every data item $\mathbf{d} \in D$ is a scalar value corresponding to the trace/sublog $i$. If we, however, consider sets of activities or activity pairs, then each data item is a vector. Therefore, we need to consider both univariate and multivariate hypothesis tests.

We will use the univariate two-sample Kolmogorov–Smirnov test (KStest) and Mann–Whitney $U$ test (MWtest) as hypothesis tests for univariate data, and the two sample Hotelling $T^2$ test for multivariate data. The KStest evaluates the hypothesis "Do the two independent samples represent two different cumulative frequency distributions?" whereas the MWtest evaluates the hypothesis "Do the two independent samples have different distributions with respect to the rank ordering of the values?". The multivariate Hotelling $T^2$ test is a generalization of the $t$-test and evaluates the hypothesis "Do the two samples have the same mean pattern?". All of these tests yield a significance probability assessing the validity of the hypothesis on the samples. We refer [43] for a classic introduction to various hypothesis tests.

## VI. FRAMEWORK

We propose the framework shown in Fig. 8 for analyzing concept drifts in process mining. The framework identifies the following steps:

*1) Feature extraction and selection:*
This step pertains in defining the characteristics of the traces in an event log. In this paper, we have defined four features that characterize the control-flow perspective of process instances in an event log. Depending on the focus of analysis, we may define additional features, e.g., if we are interested in analyzing changes in organizational/resource perspective, we may consider features derived from social networks as a means of characterizing the event log. In addition to feature extraction, this step also involves feature selection. Feature selection is important when the number of features extracted is large. We may consider dimensionality reduction techniques [44], [45] such as PCA [46] or random projection [47] to deal with high dimensionality.

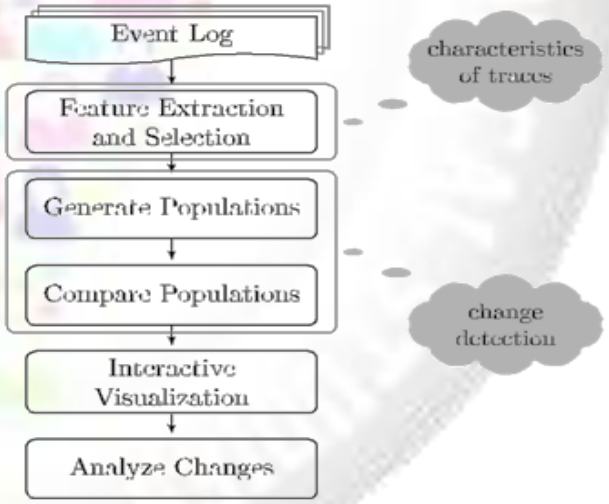



Fig.8 : Framework for handling concept drifts in process mining.

*2) Generate populations:*
An event log can be transformed into a data stream based on the features selected in the previous step. This step deals with defining the sample populations for studying the changes in the characteristics of traces. Different criteria/scenarios may be considered for generating these populations from the data stream. In Section VII, we have considered nonoverlapping, continuous, and fixed-size windows for defining the populations. We may also consider, for example, noncontinuous windows (there is a gap between two populations), adaptive windows (windows

canbeofdifferent lengths) [16],andsoon,whichare moreappropriatefordealingwithgradualandrecurring drifts.

*3) Comparepopulations:*

Oncethesamplepopulations are generated, thenextstepistoanalyzethesepopulations forany changeincharacteristics.Inthis paper,weadvocatetheuseofstatisticalhypothesistestsforcompa ring populations.The nullhypothesisinstatisticaltestsstates thatdistributions (ormeans,orstandarddeviations)of thetwosamplepopulations areequal.Dependingon desiredassumptions andthefocusofanalysis,different statisticaltestscanbeused.

*4) Interactive visualization:*

The results of comparative studiesonthepopulationsoftracecharacteristicscan beintuitivelypresented toananalyst.Forexample, the significance probabilitiesofthehypothesistestscanbe visualizedasadriftplot.Troughs insuchadriftplot signifyachangeinthesignificance probabilitythereby implyingachangeinthecharacteristicsoftraces.

*5) Analyzechanges:*

Visualizationtechniquessuchasthe driftplotcanassistinidentifyingthechangepoints.Havingidenti fiedthatachangehadtakenplace,this stepdealswithtechniquesthatassistananalystincharacterizing andlocalizing thechangeandindiscovering thechangeprocess.

Theframeworkcanbeusedfordesigningnewchangedetectiona pproaches.

## VII. IMPLEMENTATION

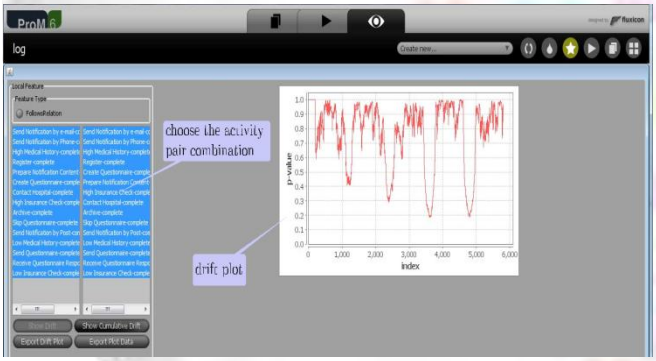



Fig.9: Visualizationofthedriftplotintheconceptdriftplug-ininProM.

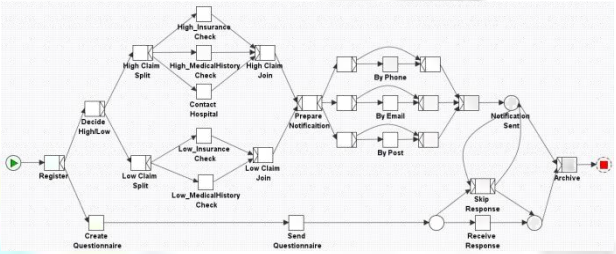
Theconceptspresentedinthispaperhavebeenrealized astheconceptdriftplug-ininthe ProM[6] framework.ProM isaplug-ableenvironment forprocessminingenvisionedto provide acommonbasisforallkindsofprocessminingtech-niquesrangingfromimporting,exporting,andfiltering event logs(process models)toanalysisandvisualization ofresults. Overyears,ProMhasemergedtobethedefactostandardfor processmining.Theconceptdriftplug-inimplementsallofthe stepsintheproposedframework andcanbeeasilyextended with additionalelements(e.g.,new featurescan be easily added).Theplug-insupports visualization ofthesignificance probabilityforthehypothesistests asadriftplot.Fig. 9shows adriftplotfromtheplug-in.

## VIII. EXPERIMENTAL RESULTS AND DISCUSSION

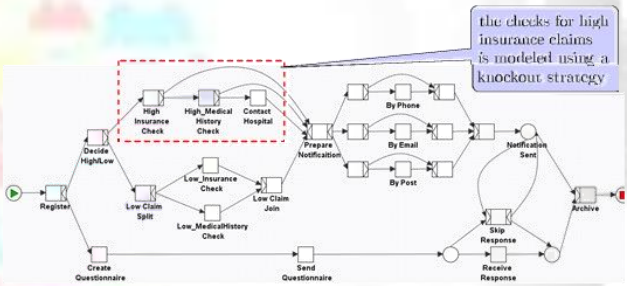
Now, we put the ideas proposed for handling concept driftsinpractice.Initially,weillustratetheeffectivenessof the proposedapproachesusing a synthetic exampleof an insurance claimprocessandlaterdiscussttheresultsfroma real-lifecasestudyinalargeDutchmunicipality.

### A. SyntheticLog-InsuranceClaimProcess

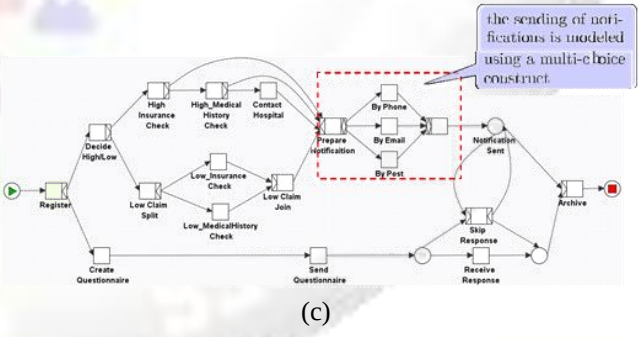
Thisprocesscorrespondstothehandlingofhealthinsurance claimsin a travelagency.Uponregistrationof a claim,a generalquestionnaire issenttotheclaimant.Inparallel,a registeredclaimisclassified ashighorlow.Forlowclaims, twoindependent tasks:1)checkinsuranceand2)check medicalhistoryneedtobeexecuted. Forhighclaims,three tasksneedtobeexecuted: 1)checkinsurance; 2)check medicalhistory;and3)contactdoctor/hospitalfor verification. Ifoneofthechecksshowsthattheclaimisnotvalid,then theclaimisrejected;otherwise,itisaccepted. Achequeand acceptancedecisionletteris preparedin caseswhereaclaimis acceptedwhilea rejectiondecisionletteriscreatedforrejected claims.Inbothcases,anotificationissenttotheclaimant.



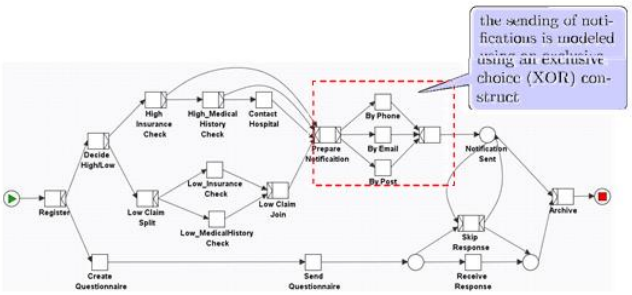

(a)



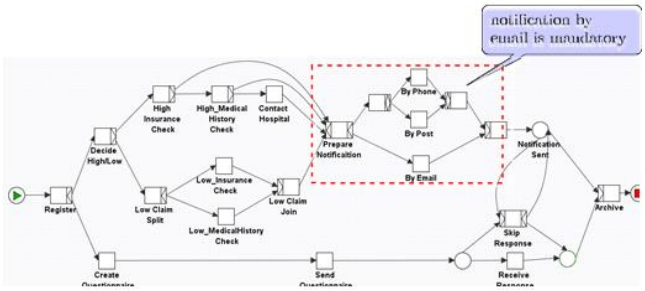

(b)




(c)




(d)




(e)

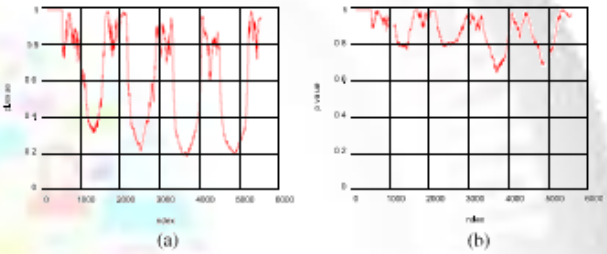
Fig. 10 : Variants of an insurance claim process of a travel agency represented in YAWL notation. Dashed rectangles: regions of change from its previous model. (a) Model 1. (b) Model 2. (c) Model 3. (d) Model 4. (e) Model 5.

Three modes of notification are supported by: 1) email; 2)telephone (fax);and3)postalmail.Thecaseshouldbe archiveduponnotifyingtheclaimant.Thiscanbedonewith orwithouttheresponse forthequestionnaire.Thedecisionof ignoringthequestionnaire,however,canonlybemadeafter anotification issent.Thecaseiscloseduponcompletionof archivingtask.Fig. 10 shows fivevariants of this process represented in YAWL [48] notation. Dashed rectangles:a change has beendoneintheprocess modelwithrespecttoitsprevious variant.Thechangescanhavevariousreasons.Forexample, inFig.10(a),thedifferentchecksforhighinsuranceclaimsare modeled usingaparallel(AND)construct.Aclaim,however, canberejectedifanyoneofthechecksfail.Insuchcases, the time and resources spent on other checks go waste. Tooptimizethisprocess,theagencycandecidetoenforce an order on these checks and proceed on checks only if theprevious checkresultsarepositive. Inotherwords,the processismodifiedwitha*knockout*strategy[49]adoptedfor theprocessfragment involvingthedifferent checksforhigh insuranceclaims, asshowninFig.10(b).Asanother example, theOR-constructpertainingtothesendingofnotification to claimantsinFig.10(c)hasbeenmodified toanexclusive-or (XOR)constructinFig.10(d).Theorganization couldhave takenadecisiontoreducetheirworkforce asacost-cutting measure. Because of the availability of limited resources, theywouldliketominimizetheredundancy ofsendingthe notification throughdifferentmodesofcommunicationand restrictittoonlyoneofthemodes.Consideringanevent logcontaining casesthatbelongtosuchamixofprocess variants,theobjective ofchangepointdetectionistodetect when theprocesseshavechanged.Inthissection,weillustrate thehandling ofconceptdriftsinthecontextofsuddenand gradualdrifts.Wehavemodeledeachofthesefive process variants inCPNtools[50]andsimulated 1200tracesforeach model.

*1) Sudden DriftChange (Point)Detection:*

Tosimulatethe suddendriftphenomenon,wecreatedaneventlog$L$consisting of6000tracesbyjuxtaposingeachsetofthe1200traces.The eventlogcontains15activitiesoreventclasses(i.e.,$|A|$ =15) and58783events(whichisthetotalnumberofeventsinthe logforallthetraces).Giventhiseventlog$L$,ourfirstobjective istodetectthefourchangepointspertaining tothesefive processvariants,asshowninFig.11(a).Globalfeaturescan beappliedonlyattheloglevel;tofacilitatethis,wehave splitthelogint0120sublogsusingasplitsizeof50traces.

Inthisscenario,thefourchangepointscorresponding tothe five processvariantsare,asshowninFig.11(b).Wehave computedthefollowsRCofall15activitiestherebygenerating amultivariate vectorof45featuresforeachsublog.Wehave appliedtheHotelling$T^2$ hypothesistestonthismultivariate datasetusingamovingwindowpopulationofsize,$w$=10. Forthishypothesistest,wehaverandomlychosen12ofthe 45featureswitha10-fold crossvalidation.[7]Fig.12(a)shows theaveragesignificanceprobabilityoftheHotelling$T^2$ test forthe10foldsonthisfeatureset.Thetroughs intheplot signifythatthereisachangeinthedistributionofthefeature valuesinthelog.In otherwords,theyshowthat thereisadrift (change)intheconcept,whichherecorrespondstotheprocess. Itisinterestingto observethatthetroughsareobservedaround indexes 24,72,and96whichareindeedthepointsofchange (rememberthat,wehavesplitthelogint0120sublogswith thechangepointsatindexes24,48,72,and96).Thechangeatinde x 48correspondingtothetransitionfrom$M_2$to$M$ could not be uncoveredusing this featuresetbecausetheRCswouldbealikeforlogsgeneratedfrom thesetwoprocessvariants.



(a) (b)

Fig.13: Averagesignificanceprobability(overallactivitypairs)ofKStest ontheJmeasureandWCfeaturesetsestimatedforeachtrace. *x*-axis:trace index. *y*-axis:significanceprobabilityofthetest. Troughs:changepoints. Verticalgridlines:actual(sudden)changepoints.(a)J-measure.(b)WC.

correspondingtoeach activity pairin$A\times A$.Fig.13(a)shows theaveragesignificance probabilityofKStestonallactivity pairs,whereasFig.13(b)showstheaverage significance probabilityofKStestonallactivitypairsusingtheWCfeature set.Wecanobservethatsignificant troughsareformedat indexes 1200,2400,3600,and4800.Theseareindeedthe pointswherethemodelshavebeenchanged.

- *Influence of Population Size:*

Itisimperativetonotethat thegoodnessoftheresultsofhypothesistestsdependson thepopulation size.Thestatisticalanalysisassumesthateach population isindependent. Agoodpopulation sizeislargely dependent ontheapplicationandthefocusofanalysis.To studytheinfluenceofpopulationsize,wehaveconsideredthe $J$ measureforeverypairofactivitiesandtheunivariateKS testforchange pointdetection.Fig.14showstheresultsfor varyingsizesofthepopulation.Weobservealotofnoise forsmallpopulations andthedrifttendstobesmoothasthe populationsizeincreases.Thiscanbeattributedtothefact thatasthepopulationsizeincreases(i.e.,asweconsidermore cases),thevariabilityinthenatureofcasesreducesandattains astability,e.g.,therecanbeafluxoflow-insuranceclaims initiallyandafteracertaintimetheproportionstabilizes.

*2) SuddenDrift ChangeLocalization:*

Oursecondobjective inhandling concept driftsisthatofchangelocalization. To localizethechanges (identifytheregions ofchange),weneed to consider activity pairs individually or subsets of activitypairs. For example, the change from M1 to M2 is localized in the region pertaining to high insurance claim checks. We expect characteristic changes in features pertaining to these activities and other activities related to these activities. For example, in M1 , the activities High Medical History Check and Contact Hospital always follow the activity Register whenever a claim is classified as high. In contrast, in M2 , these activities need not always follow Register because both these activities are skipped if High Insurance Check fails while Contact Hospital is skipped if High Medical History Check fails. During simulation, we have set the probability of success of a check to 90%. We have considered the WC feature for the activity relation Contact Hospital follows Register on a window length of l = 10 in each trace separately. Fig. 15(a) shows the significance probability of the univariate KS test using a population size of w = 400 on this feature. We can observe that one dominant trough is formed at index 1200 showing that there exists a change in the region between Register and Contact Hospital. No subsequent changes with respect to this activity pair can be noticed, which is indeed the case in the sequence of models used.

the accuracy of the proposed framework in handling gradual drifts. Recall that in gradual drifts, one concept fades gradually while the other takes over. This phenomenon of gradual change can be modeled in many ways. In this paper, we consider the scenario where the change is linear between two sources, as shown in Fig. 16(a). In this figure, we observe the fading of one concept M1 and the taking over of another concept M2 happen linearly. Within this setup, we can alter the extent to which the two concepts coexist. For the insurance claim example, we generated two event logs exhibiting gradual drifts by varying the duration of change. In the first case, the process variants M1 and M2 coexist between trace indexes 1000 and 1400, the variants M2 and M3 coexist between indexes 2200 and 2600, the variants M3 and M4 coexist between indexes 3400 and 3800, and the variants M4 and M5 coexist between indexes 4600 and 5000, as shown in Fig. 16(b). The point of cross over is still retained at indexes 1200, 2400, 3600, and 4800.

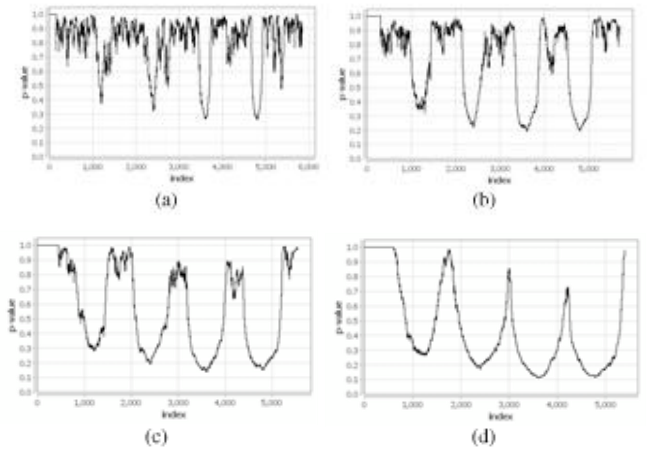



Fig. 14 : Average significance probability (over all activity pairs) for different population sizes of KS test on the J measure estimated over all activity pairs in each trace. x - axis: trace index. y-axis: significance probability of the test. Troughs: change points. Vertical grid lines: actual (sudden) change points. (a) w = 150, (b) w = 300, (c) w = 450, and (d) w = 600.

*B. Real-LifeLog:AdvertisementPermitProcess ofaDutchMunicipality*

Thesyntheticeventdatacreatedthroughsimulationallow ustocomparethecontrolled (ground truth)changeswiththe detectedchanges.We,however,havealsoapplied ourconcept driftanalysistechniques tovariousreal-lifeeventlogs.Here, wereportonacasestudywhereweanalyzedconceptdrifts inprocesseswithinalargeDutchmunicipality.Municipalities areinterestedinobtaininginsightsintotheirprocesses,e.g., thewaytheyareplannedtobeexecutedvis-a-vistheway

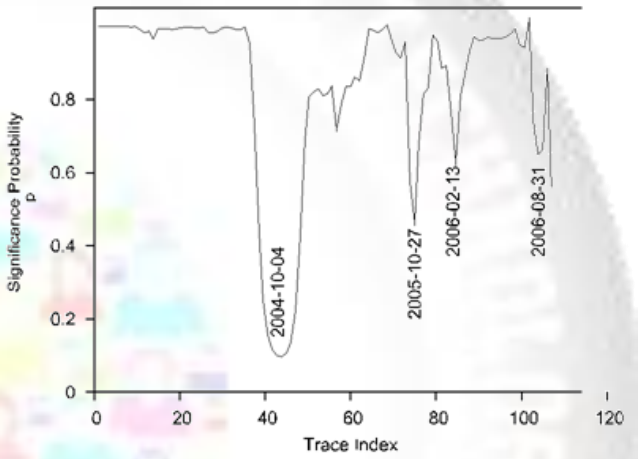



Fig.19. Averagesignificanceprobability(overallactivitypairs)ofKStest on *J* measure.ThepopulationsizefortheKStestis10.Therearefour troughssignifyingachangeinbehavior.

follows.

1) Upon submissionof an application,the municipality acknowledgesthereceiptofdocumentsand(optionally) testsforitscompleteness.

2) Then, the municipality proceeds with a follow-up procedurethatverifies whethertheapplicationand submitteddocuments areincompliance withthe regulations.

3) Withtheinvestigations, thenthemunicipalitymakesa decisionontheapplication andinformstheapplicant withthedecisionalongwithafeeletter.

4)Finally,themunicipality registerstheadvertisements placedandenforcesthem.

Fig.20(b)showstheprocessmodeldiscoveredusingthe Heuristicminer[56]on theeventlog*L* Thefigurehighlights regionsthatdiffer from theprocessmodelinFig.20(a).There aretwochangesin thismodelwithrespectto thepreviousone. Thefirstchangeisrelatedtothecheckingforcompleteness of the registered documents. In the initial process model [Fig.20(a)],this checkwasnot mandatory(onlytwoofthe43 applicationswerecheckedforcompleteness).Themunicipality changedthisprocessbymakingthechecksmandatory before proceeding. Thesecondchangeistheintroduction ofanew

activity‘Endprocedure;enforcementisnext,’asshownin Fig.20(b).Theinitialprocessmodelhadonlytheactivity ‘Endprocedure,possiblychooseenforcement’ whereasthe newmodelhasboththeseactivities.Similarchangeshave beenobserved intherestofthemodels.Wedonotprovide themherebecauseofspaceconstraints. Foramoredetailed discussiononthiscasestudyrefer[57].

### *C. TimeComplexity*

Inthissection,weassessthetimecomplexity ofthepro-posedapproach.Thefeatureextractionisdependentonthesize oftheeventlog(i.e.,thenumberofevents).Thefeaturevalues forallofthefeaturesproposedinthispapercanbeextractedin lineartimewithrespecttothesizeoftheevent log.Fig.21(a) showstheaveragetimealongwith95%confidence intervals (overfiveindependentruns) for extractingthe *J* measure

## IX. Conclusion

In this paper, we have introduced the topic of concept drift in process mining, i.e., analyzing process changes based on event logs. We proposed feature sets and techniques to effectively detect the changes in event logs and identify the regions of change in a process. Our initial results show that heterogeneity of cases arising because of process changes can be effectively dealt with by detecting concept drifts. Once change points are identified, the event log can be partitioned and analyzed. This is the first step in the direction of dealing with changes in any process monitoring and analysis efforts. We have considered changes only with respect to the control- flow perspective manifested as sudden and gradual drifts. Therefore, our analysis should only be observed as the starting point for a new subfield in the process mining domain and there are lots of challenges that still need to be addressed. Some of these challenges include.

1) Change-pattern specific features:
2) Feature selection:
3) Recurring drifts:
4) Holistic approaches:
5) Change process discovery:
6) Sample complexity:
7) Online (on-the-fly) drift detection:

## References

[1] (2010). All-in-one Permit for Physical Aspects: (Omgev- ingsvergunning) in a Nutshell [Online]. Available: http://www. answersforbusiness.nl/regulation/all-in-one-permit-physical-aspects

[2] United States Code. (2002, Jul.). Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745 [Online]. Available: http://files.findlaw. com/news.findlaw.com/cnn/docs/gwbush/sarbanesoxley072302.pdf

[3] W. M. P. van der Aalst, M. Rosemann, and M. Dumas, “Deadline-based escalation in process-aware information systems,” Decision Support

[4]SyMst..,Dvuoml. a4s3, ,Wn.oM. 2. ,Pp. pv.an49d2e–r 5A1a1l,st2, 0a1n1d. A. H. M. Ter Hofstede, Process-Aware Information Systems: Bridging People and Software Through Process Technology. New York, NY, USA: Wiley, 2005.

[5] W. M. P. van der Aalst and K. M. van Hee, Workflow Management: Models, Methods, and Systems. Cambridge, MA, USA: MIT Press,2004.

[6] W. M. P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes. New York, NY, USA: Springer- Verlag, 2011.

[7] B. F. van Dongen and W. M. P. van der Aalst, “A meta model for process mining data,” in Proc. CAiSE Workshops (EMOI-INTEROP Workshop), vol. 2. 2005, pp. 309–320.

[8] C. W. Günther, (2009). XES Standard Definition [Onlilne]. Available: http://www.xes-standard.org

[9] F. Daniel, S. Dustdar, and K. Barkaoui, “Process mining manifesto,” in BPM 2011 Workshops, vol. 99. New York, NY, USA: Springer-Verlag, 2011, pp. 169–194.

[10] R. P. J. C. Bose, W. M. P. van der Aalst, I. Žliobaitė, and M. Pechenizkiy,“Handling concept drift in process mining,” in Proc. Int. CAiSE, 2011 pp. 391–405

[11] J. Carmona and R. Gavaldà, “Online techniques for dealing with concept drift in process mining,” in Proc. Int. Conf. IDA, 2012, pp. 90–102.

[12] J. Schlimmer and R. Granger, “Beyond incremental processing: Tracking concept drift,” in Proc. 15th Nat. Conf. Artif. Intell., vol. 1. 1986, pp. 502–507.

[13] A. Bifet and R. Kirkby. (2011). Data Stream Mining: A Practical Approach, University of Waikato, Waikato, New Zealand [Online]. Available: http://www.cs.waikato.ac.nz/~abifet/MOA/StreamMining.pdf

[14] I. Žliobaitė, “Learning under concept drift: An Overview,” CoRR, vol. abs/1010.4784,2010[Online]. Available: http://arxiv.org/abs/1010.4784

[15] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in Proc. SBIA, 2004, pp. 286–295.

[16] A. Bifet and R. Gavaldà, “Learning from time-changing data with adaptive windowing,” in Proc. 7th SIAM Int. Conf. Data Mining (SDM),2007, pp. 443–448.

[17] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, “Exponentially weighted moving average charts for detecting concept drift,” Pattern Recognit. Lett., vol. 33, no. 2, pp. 191–198, 2012.

[18] K. Nishida and K. Yamauchi, “Detecting concept drift using statistical testing,” in Proc. 10th Int. Conf. Discovery Sci.. 2007, pp. 264–269.

[19] C. Alippi and M. Roveri, “Just-in-time adaptive classifiers–Part I: Detecting nonstationary changes,” IEEE Trans. Neural Netw., vol. 19,no. 7, pp. 1145–1153, Jul. 2008.

[20] C. Alippi, G. Boracchi, and M. Roveri, “Just-in-time classifiers for recurrent concepts,” IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 4, pp. 620–634, Apr. 2013.

[21] J. Z. Kolter and M. A. Maloof, “Dynamic weighted majority: An ensemble method for drifting concepts,” J. Mach. Learn. Res., vol. 8, pp. 2755–2790, Jan. 2007.

[22] R. Elwell and R. Polikar, “Incremental learning of concept drift in nonstationary environments,” IEEE Trans. Neural Netw., vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[23] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," Mach. Learn., vol. 23, no. 1, pp. 69–101,Apr. 1996.
[24] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," Knowl. Based Syst., vol. 18, nos. 4–5, pp. 187–195, Aug. 2005.
[25] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Han- dling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections," in Proc. 19th IEEE Int. Symp. CBMS, Nov. 2006, pp. 679–684.
[26] M. Pechenizkiy, J. Bakker, I. Žliobaitė, A. Ivannikov, and T. Kärkkäinen, "Online mass flow prediction in CFB boilers with explicit detec- tion of sudden concept drift," SIGKDD Explorations, vol. 11, no. 2, pp. 109–116, 2009.
[27] E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, nos. 1–2, pp. 100–115, 1954.
[28] N. Mulyar, "Patterns for process-aware information systems: An approach based on colored Petri nets," Ph.D. dissertation, Dept. Comput. Sci., Univ. Technol., Eindhoven, The Netherlands, 2009.
[29] B. Weber, S. Rinderle, and M. Reichert, "Change patterns and change support features in process-aware information systems," in Proc. 19th Int., 2007, pp. 574–588.
[30] G. Regev, P. Soffer, and R. Schmidt, "Taxonomy of flexibility in business processes," in Proc. 7th Workshop BPMDS, 2006, pp. 1–4.
[31] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. M. P. van der Aalst, "Process flexibility: A survey of contemporary approaches," in Proc. Adv. Enterprise Eng. I, 2008, pp. 16–30.
[32] K. Ploesser, J. C. Recker, and M. Rosemann, "Towards a classification and lifecycle of business process change," in Proc. BPMDS, vol. 8. 2008, pp. 1–9.
[33] C. W. Günther, S. Rinderle-Ma, M. Reichert, and W. M. P. van der Aalst, "Using process mining to learn from process changes in evolu- tionary systems," Int. J. Business Process Integr. Manag., vol. 3, no. 1, pp. 61–78, 2008.
[34] M. van Leeuwen and A. Siebes, "StreamKrimp: Detecting change in data streams," in Proc. Mach. Learn. Knowl. Discovery Databases, 2008, pp. 672–687.
[35] I. Žliobaitė and M. Pechenizkiy. (2010). Handling Concept Drift in Information Systems [Online]. Available:http://sites.google.com/site/zliobaite/CD_applications_2010.pdf
[36] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining. 2003, pp. 226–235.
[37] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," IEEE Trans. Neural Netw. Learn. Syst., Apr. 2013, doi: 10.1109/TNNLS.2013.2251352.
[38] W. M. P. van der Aalst, A. Adriansyah, and B. Dongen, "Replay- ing history on process models for conformance checking and perfor- mance analysis," WIREs Data Mining Knowl. Discovery, vol. 2, no. 2, pp. 182–192, 2012.
[39] M. Hammer, Beyond Reengineering: How the Process-Centered Orga- nization is Changing Our Work and Our Lives. New York, NY, USA: Harper business, 1996.
[40] L. L. Minku, A. P. White, and X. Yao, "The impact of diver- sity on online ensemble learning in the presence of concept drift," IEEE Trans. Knowl. Data Eng., vol. 22, no. 5, pp. 730–742, May 2010.
[41] P. Smyth and R. M. Goodman, Rule Induction Using Information Theory. Washington, DC, USA: AAAS Press, 1991, pp. 159–176.
[42] N. M. Blachman, "The amount of information that y gives about X ,"IEEE Trans. Inf. Theory, vol. 14, no. 1, pp. 27–31, Jan. 1968.
[43] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures. London, U.K.: Chapman & Hall/CRC, 2004.
[44] I. K. Fodor, "A survey of dimensionality reduction techniques," in Proc. Center Appl. Sci. Comput., Lawrence Livermore Nat. Lab., 2002, pp. 1–24.
[45] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Mar. 2003.