ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

# Dealing With Concept Drifts in Process Mining

**Ms.P.Aruna**
M.Tech Student,
Dept of CSE,
KITS for Women's, Kodad, T.S, India.

**Ms.P. Laxmi Priya**
Associate Professor,
Dept of CSE,
KITS for Women's, Kodad, T.S, India.

## Abstract:

Concept drift is an important concern for any data analysis scenario involving temporally ordered data. In the last decade Process mining arose as a discipline that uses the logs of information systems in order to mine, analyze and enhance the process dimension. There is very little work dealing with concept drift in process mining. In this paper we present the _rst online mechanism for detecting and managing concept drift, which is based on abstract interpretation and sequential sampling, together with recent learning techniques on data streams. 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 drifts and real-life event data from a Dutch municipality.

**Index Terms:** Concept drift, flexibility, hypothesis tests, process changes, process mining.

## 1 Introduction:

In recent years process mining techniques have matured. Provided that the pro- cess 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 perfor- mance analysis, compliance checking, and prediction. Unfortunately, most pro- cesses 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 exible 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 e_ects, natural calamities and disasters, deadline esca- lations [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-ow, 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 _rst and most fundamental problem is to de- tect 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 organiza- tion (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 identi_ed, 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 identi_cation of change perspective (e.g., control-ow, data, resource, sudden, gradual, etc.) and the identi_cation 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 o_ers are provided during holiday seasons. 3. Change Process Discovery: Having identi_ed, 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 sea- sonal 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 met- rics (service levels, throughput time, etc.) of a process at di_erent instances of time. One can consider an event log L as a time series of traces (traces ordered based on the timestamp of the _rst event). The basic premise in handling concept drifts is that the characteristics of the traces before the change point di_er 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-ow perspective of a process characterizes the relationships be- tween activities. Dependencies between activities in an event log can be cap- 2 tured and expressed using the follows (or precedes) relationship, also referred to as causal footprints. Bose et al. [4] proposed four features characterizing the control-ow dependencies between activities. These features are shown to be e_ective in detecting process changes. An event log can be transformed into a data set 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 populations1 of feature values (of some population size w, see Fig. 1) and inves- tigating if there is a signi_cant di_erence between two successive populations. The premise is that di_erences are expected to be perceived at change points provided appropriate characteristics of the change are captured as features. The di_erence between populations is assessed using statistical hypothesis testing [5]. Hypothesis tests yield a signi_cance 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 signi_cant di_erences (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.

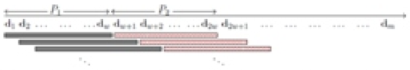

$P_1$
$P_2$
$d_1$ $d_2$ … … …$d_w$ $d_{w+1}$ $d_{w+2}$ … …$d_{2w}$ $d_{2w+1}$ … … … … … $d_m$

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 classi_ed into on- line and o_ine depending on whether or not the presence of changes or the occurrence of drifts needs to be uncovered in real-time.

## 3. RELATED WORK:

Over the last two decades many researchers have been working on process flexibility, e.g., making workflow systems adaptive. In [28] and [29] collections of typical change patterns are described. In [30] and [31] extensive taxonomies of the various flexibility approaches and mechanisms are provided.Ploesser et al. [32] have classified business process changes into three broad categories: 1) sudden; 2) anticipatory; and 3) evolutionary. This classification is used in this paper, but now in the context of event logs. Despite the many publications on flexibility, most process mining techniques assume a process to be in a steady state. A notable exception is the approach in [33]. This approach uses process mining to provide an aggregated overview of all changes that have happened so far. This approach, however, assumes that change logs are available, i.e., modifications of the workflow model are recorded. At this point of time, very few information systems provide such change logs. Therefore, this paper focuses on concept drift in process mining assuming only an event log as input. Furthermore, the tool support provided by the authors has some limitations in its applicability.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The tool does not detect change points and does not work on logs with multiple process changes, i.e., it does not detect the presence/absence of multiple changes and does not report when (the trace index) process changes have happened. The tool just reports that a change exists and terminates (if changes exist) and does not terminate if no changes exist. In contrast, our tool can handle multiple process changes and can detect both the presence of and the points of change in addition to being able to assist in change localization.

## 4. 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.

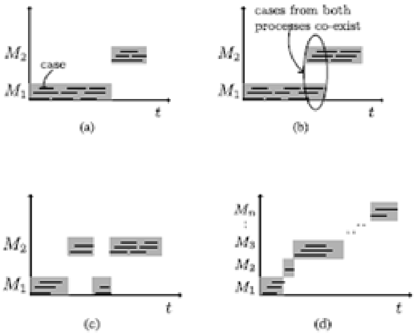
### A. Perspectives of Change:

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:

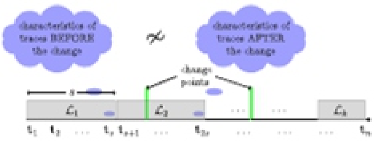
This class of changes deals with the behavioral and structural changes in a process model. Just like the design patterns in



Control-flow process discovery:
(a) log containing a drift from trace 8 on,
(b)Petri net discovered from part of the log

## V. 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 is the set of activities. A+ is the set of all nonempty finite sequences of activities from A.  2) A process instance (i.e., case) is described as a trace over A, i.e., a finite sequence of activities. Examples of traces are abcd and abbbad. 3) Let t = t(1)t(2)t(3) . . . t(n)  A+ be a trace over A. |t| = n is the length of the trace t. t(k) is the kth activity in the trace and t(i, j ) is the continuous subsequence of t that starts at position i and ends at position j . ti = t(i, |t|) represents the suffix of t that begins at position i . 4) An event log, L, corresponds to a multiset (or bag) of traces from A+. For example, L = [abcd, abcd, abbbad] is a log consisting of three cases. Two cases follow trace abcd and one case follows trace abbbad. 5) N,N0, and R + 0 are the set of all natural numbers, the set of all natural numbers including zero, and the set of all positive real numbers including zero, respectively.



Event log visualized as a time series of traces along with change points. The basic premise of change (point) detection is that characteristic differences exist in the traces before and after the change. of the traces after the change point. The problem of changepoint detection is then to identify the points in time where the process has changed, if any. Change point detection involves two primary steps:
1) capturing the characteristics of the traces;
2) identifying when the characteristics change.

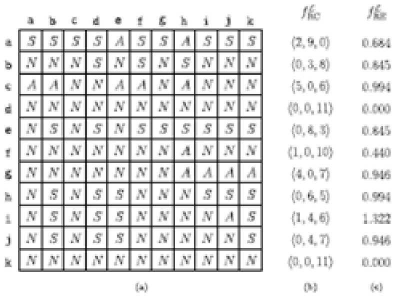
We refer to the former step as feature extraction and the latter step as drift detection. The characteristics of the traces can either be defined for each trace separately or can be done at a sublog level. An event log can be split into sublogs of s traces (s  N is the split size). We can consider either overlapping or nonoverlapping sliding windows when creating such sublogs. Fig. 4 shows the scenario where two subsequent sublogs do not overlap. In this case, we have k = _ns _ sublogs for an event log of n traces.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

Thus, the logs processed to determine the characteristics of traces can be observed as a data stream of feature values where statistical tests can be used to detect changes. As mentioned earlier, dealing with concept drifts in processmining involves two primary steps. First, we need to capturethe characteristics of traces; we propose a few feature sets that address this in Section VI. Second, we need to identify when these characteristics change; we look at techniques that address this in Section VII.

## 5. FEATURE EXTRACTION:

Event logs are characterized by 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. For any pair of activities a, b  A, and a trace t = t(1)t(2) t(3) . . . t(n)  A+, we say b follows a if and only if for all 1 ≤ i ≤ n such that t(i ) = a there exists a j such that i < j ≤ n and t( j ) = b. In temporal logic notation: _(a  (♦b)). We say a precedes b if and only if for all 1 ≤ j ≤ n such that t( j ) = b there exists an i such that 1 ≤ i < j and t(i ) = a, i.e., aWb where W is the weak until in linear temporal logic notation. The follows and precedes relationships can be lifted from traces to logs. If b follows a in all the traces in an event log, then we say that b always follows a. If b follows a only in some subset of the traces, then we say that b sometimes follows a. If b does not follow a in all traces,then we say that b never follows a. Consider an event log L = [acaebfh, ahijebd, aeghijk] containing three traces defined over A ={a, b, c, d, e, f, g, h, i, j, k}. The following relations hold in L: e always follows a, e never follows b, and b sometimes follows a. Fig. 5(a) shows the relationship between.



| | a | b | c | d | e | f | g | h | i | j | k | $f_{RC}$ | $f_{RE}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | S | S | S | S | A | S | S | A | S | S | S | (2, 9, 0) | 0.684 |
| b | N | N | N | S | N | S | N | S | N | N | N | (0, 3, 8) | 0.845 |
| c | A | A | N | N | A | A | N | A | N | N | N | (5, 0, 6) | 0.994 |
| d | N | N | N | N | N | N | N | N | N | N | N | (0, 0, 11) | 0.000 |
| e | N | S | N | S | N | S | S | S | S | S | S | (0, 8, 3) | 0.845 |
| f | N | N | N | N | N | N | N | A | N | N | N | (1, 0, 10) | 0.440 |
| g | N | N | N | N | N | N | N | A | A | A | A | (4, 0, 7) | 0.946 |
| h | N | S | N | S | S | N | N | N | S | S | S | (0, 6, 5) | 0.994 |
| i | N | S | N | S | S | N | N | N | N | A | S | (1, 4, 6) | 1.322 |
| j | N | S | N | S | S | N | N | N | N | N | S | (0, 4, 7) | 0.946 |
| k | N | N | N | N | N | N | N | N | N | N | N | (0, 0, 11) | 0.000 |
| | (a) | | | | | | | | | | | (b) | (c) |

Feature extraction (a) causal footprint matrix for all activity pairs (b) relation type count (RC) and (c) relation entropy (RE) feature values. A: always follows, N: never follows, and S: sometimes follows.

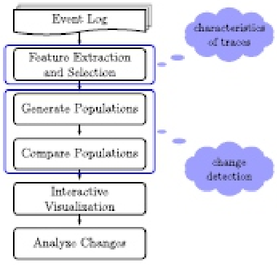
6.HYPOTHESIS TESTS FOR DRIFT DETECTION

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 di □ 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.

## 7. 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


Event Log
characteristics of traces
Feature Extraction and Selection
Generate Populations
Compare Populations
change detection
Interactive Visualization
Analyze Changes

Framework for handling concept drifts in process mining.

## 8. IMPLEMENTATION:

The concepts presented in this paper have been realized as the concept drift plug-in in the ProM6 framework. ProM is a plug-able environment for process mining envisioned to provide a common basis for all kinds of process mining techniques ranging from importing, exporting, and filtering event logs (process models) to analysis and visualization of results.Over years, ProM has emerged to be the de facto standard forprocess mining. The concept drift plug-in implements all of the steps in the proposed framework and can be easily extended with additional elements (e.g., new features can be easily added). The plug-in supports visualization of the significance probability for the hypothesis tests as a drift plot. Fig. 9 shows a drift plot from the plug-in.

## 9. EXPERIMENTAL RESULTS AND DISCUSSION:

Now, we put the ideas proposed for handling concept drifts in practice. Initially, we illustrate the effectiveness of the proposed approaches using a synthetic example of an insurance claim process and later discuss the results from a real-life case study in a large Dutch municipality. A. Synthetic Log-Insurance Claim Process This process corresponds to the handling of health insurance claims in a travel agency. Upon registration of a claim, a general questionnaire is sent to the claimant. In parallel, a registered claim is classified as high or low. For low claims, two independent tasks: 1) check insurance and 2) checkm edical history need to be executed. For high claims, three tasks need to be executed: 1) check insurance; 2) check medical history; and 3) contact doctor/hospital for verification. If one of the checks shows that the claim is not valid, then the claim is rejected; otherwise, it is accepted. A cheque and acceptance decision letter is prepared in cases where a claim is accepted while a rejection decision letter is created for rejected claims. In both cases, a notification is sent to the claimant.

## 10. 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 controlflow 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: In this paper, we presented very generic features (based on follows/precedes relation). These features are neither complete nor sufficient to detect all classes of changes. An important direction of research would be to define features catering to different classes of changes and investigate their effectiveness. A taxonomy/classification of change patterns and the appropriate features for detecting changes with respect to those patterns are needed. 2) Feature selection: The feature sets presented in this paper result in a large number of features. For example, the activity relation count feature type generates 3×|A|features whereas the WC and J measure generate |A|2features (corresponding to all activity pairs). On the one hand, such high dimensionality makes analysisintractable for most real-life logs. On the other hand,changes being typically concentrated in a small region of a process make it unnecessary to consider all features. There is a need for tailored dimensionality reduction techniques [44], [45] that can efficiently select the most appropriate features. 3) Holistic approaches: In this paper, we discussed ideas on change detection and localization in the context of sudden and gradual changes to the control-flow perspective of a process. As mentioned in Section IV, the data and resource perspectives are also, however, equally important. Features and techniques that can enable the detection of changes in these other perspectives need to be discovered. Furthermore, there could be instances where more than one perspective (e.g., both control and resource) change simultaneously. Hybrid approaches considering all aspects of change holistically need to be developed. 4) Recurring drifts: When dealing with recurring drifts, in addition to change point detection and change localization, it is important to identify the variant(s) that recur. This requires robust metrics to assess the similarity between process variants and/or event logs.

## 9. ACKNOWLEDGMENTS:

I am P.ARUNA and would like to thank the publishers, researchers for making their resources material available. I am greatly thankful to Associate Prof Ms .P. LAXMI PRIYA for their guidance. We also thank the college authorities, PG coordinator and Principal for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

## 10. .REFERENCES:

[1] (2010). All-in-one Permit for Physical Aspects: (Omgevingsvergunning) 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/gw-bush/sarbanesoxley072302.pdf

[3] W. M. P. van der Aalst, M. Rosemann, and M. Dumas, "Deadline-basedescalation in process-aware information systems," Decision SupportSyst., vol. 43, no. 2, pp. 492–511, 2011.170 IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 25, NO. 1, JANUARY 2014

[4] M. Dumas, W. M. P. van der Aalst, and A. H. M. Ter Hofstede, Process-Aware Information Systems: Bridging People and Software ThroughProcess 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 andEnhancement 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 processmining 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," inBPM 2011 Workshops, vol. 99. New York, NY, USA: Springer-Verlag,2011, pp. 169–194.

## Author's Details:

**Ms.P.Aruna.** MTech student, in M.Tech Student, Dept of CSE in KITS for women's,kodad, T.S, India

**Ms.P. Laxmi priya** working as a Associate at CSE in KITS for women's,kodad, T.S, IndiaJNTUH Hyderabad. He has 2 years of UG/PG Teaching Experience