ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

CONSTRUCTION OF IMPROVED PROCESS MODELS BY CLUSTERING EVENT LOGS

Swapnali Sonawane [*1], Prof.D.S.Kulkarni [2]
Computer Engineering Department, Dr.D.Y.Patil College of Engineering, Pune, India.

DOI:

ABSTRACT

One main motive is to solve the problem that presently exist in process discovery, which includes unable to search out correct and understandable process models out of event logs stemming from exceptionally flexible environments. Programming analyst spend extra time in dealing with programming bugs. An unavoidable stride of fixing bugs is bug processing in a system, because of this to efficaciously relegate a designer to every other bug. To decrease the time cost in manual work, content classifications are linked to lead programmed bug processing. This system addresses the difficulty of statistics diminishment for bug processing in a process improvement, i.e., the way to lessen the scale and identify the character of bug facts. To conquer those problems proposed system provides an automatic way for software engineers to generate mined process from systematic event logs specification and bug reviews consist of problem fixing, operating to gain others and technical task. This system focuses on characteristics from chronicled bug information units and fabricates a prescient model for each alternative bug statistics set.

KEYWORDS: Event Log, Trace Clustering, Process Discovery, Bug Processing, Process Mining.

I. INTRODUCTION

Mining process has well used for analyzing process executions supported event logs. Numbers of different techniques performed well on structured processes, however still have issues discovering the less structured processes. This is most fascinating in domains requiring flexibility. To address this, trace clustering is used that is, the event log is split into uniform sets and for every subset a model is formed. The aim is that the extraction of knowledge from real time processes. The information that is generated throughout the execution of real time processes in information systems is employed for reconstructing process models.

These models are helpful for analyzing the processes. The procedure of processing bug is going to be bug sorting, that transfer an engineer to different bug. Programming organizations pay the larger part of their expense in managing these bugs. To decrease time and cost value of bug processing, this system addresses a way to deal with a designer to resolve the new coming bug report. During this projected approach system is doing decrease in information of a bug set which is able to reduce the scale of the data and additionally build the valuable data. This system is utilizing in process improvement and highlight alternative for verifiable bug information.

II. BASIC CONCEPTS

A] Event Log:-

The starting point for process mining is an event log. Each event in such a log refers to an activity and is associated to a particular case that is a process instance. Events that are associated to a particular case are ordered and specify the process execution. Event logs can store additional data about events. In fact, whenever possible, process mining techniques use supplementary information such as the resource (person or device) executing or initiating the activity, the event's time stamp, and other data attributes. An event log is basically a table. It contains all recorded events that relate to executed activities. Each event is mapped to a case.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

B] Process Model:-

A process model is an abstraction of the execution of a process. A single execution of a process is called process instance. They are reflected in the event log as a set of events that are associated to the same case. The sequence of recorded events in a case is called trace. The model that describes the execution of a single process instance is called process instance model. A process model abstracts from the single behavior of process instances and provides a model that defines the behavior of all instances that belong to the same process. Cases and events are characterized by classifiers and attributes. Classifiers define the distinctness of cases and events by mapping unique names to each case and event. Attributes store additional information that can be used for analysis purposes. First, analysts use process discovery techniques to extrapolate a model from an event log. They can also create this initial process model manually.

III. LITERATURE SURVEY

Ferreira et al introduces the principles of sequence cluster and presents two case studies wherever the technique is employed to get behavioral patterns in event logs. A technique that mechanically teams sequences into completely different clusters so as to spot typical activity patterns. The problem usually encountered in applies is that for processes with a high diversity of behavior solely terribly advanced models may be discovered. Grouping the traces into a lot of solid clusters and discovering separate models for every of them is one strategy to get higher models.

Medeiros introduces, a technique during which many candidate answers are measure evaluated by fitness perform that determines however consistent every solution is with the log. Current techniques have issues once mining processes that contain the presence of noise within the logs. Most of the issues happen as a result of several techniques are measure supported native data within the event log. To overcome these issues, Medeiros uses genetic algorithms to mine process models. Joachim Herbst, introduces a technique to find out a hidden Markov model (HMM) that represents the structure of the initial real time processes. Workflow management systems (WFMS) supply very little action of progress models and their alteration to ever-changing necessities. To support these activities Herbst propose associate approach that induces process models from process instances.

Wil van der Aalst defines a technique that's able to re-create a Petri-net model from the ordering relations found in an event logs. Alpha algorithmic rule will mechanically extract a Petri net that offers a short model of the behavior seen in a very set of event traces, forward the traces area unit of completed instances. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos and Prabhakar Raghavan proposed hierarchical clustering algorithmic program that, given an oversized set of execution traces of one process, separates them into clusters and finds the dependency graph on an individual basis for every cluster. Dongen propose a multistep approach. He proposes as, initial models are generated for every individual process instance. Within the final step but, these instance models are united to get an overall model for the complete information set, final step, i.e., aggregating instance graphs.

IV. PROPOSED SYSTEM

In this system, it is providing a new framework for change point detection and a means to compare clustering's, which expresses high-level overview of this system approach. By incorporating the time dimension it can discover temporal evolutions in process behavior. Changes in behavior caused by differences in case data can therefore be detected. In order to detect change points, this system looks at similarities between cases. It considers the effect of new events on a clustering of active cases.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

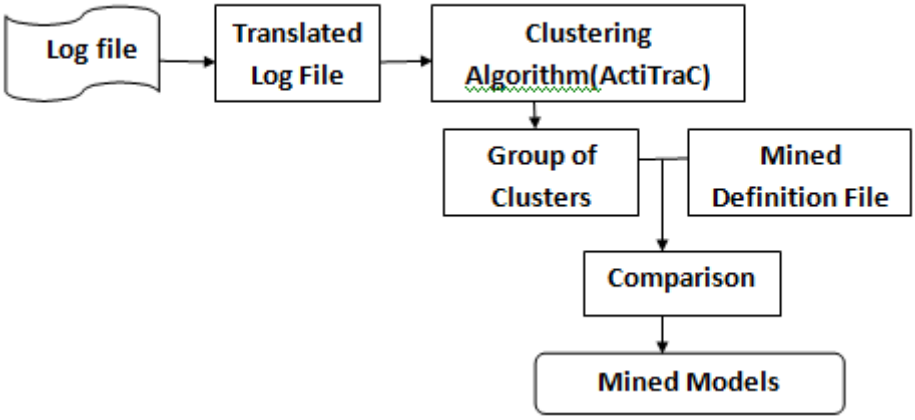

Fig 4.1System Architecture for Proposed System

This system proposes to employ the similarity matrix between cases to detect changes in behavior in traces. The occasion of new events in a particular trace will change the similarity of that trace to other trace, that is defined in the similarity matrix. As this matrix is the input for the clustering algorithm, the impact on the similarity matrix is a good indicator for how much the clustering will change. Thus, in order to detect potentially interesting change points, this system computes the change in the values of the similarity matrix over time. Similar to the approaches that use some statistical tests, a large difference indicates a significant change in behavior. Cases which have events in or before the window are measured in the calculation of the next similarity matrix. Events taking place after the current event window are not measured. A similarity matrix is calculated for that sub-log and compared with the previous similarity matrix. If change points have been identified, clustering's of cases taking place before and after these points can be created to compare behavior before and after the change point. Proposed system aims to provide an automatic way for software engineers to generate mined models from event logs specification include problem solving, working to benefit others and technical challenge.

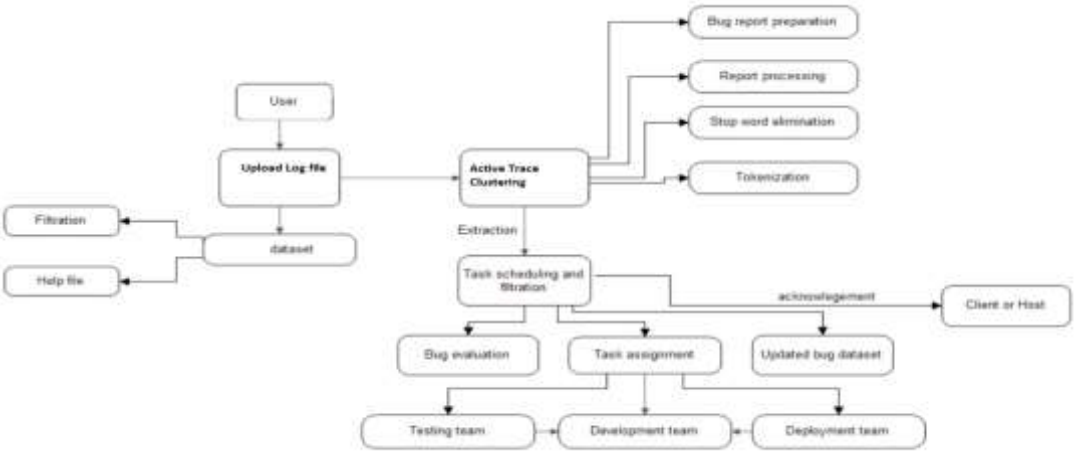

Fig.4.2: System Work Flow

This system initially first converts the event log file or bug report into some intermediate form such as translated tokenized log file and keyword filtered log file by using classifiers. Then this filtered log file format is analyzed to extract the information like similarity matrix, frequency count, most read/write data, database queries and this information is used to build the clusters. System would creates the clusters using active tracing clustering algorithm to provide distinguished description of generated models thus incorrectness and extra overhead in

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

analysis phase of model development is removed to significant extent. Proposed system uses string similarity and distance calculation method to calculate the fitness value for each trace. Also this system proposes active trace clustering algorithm for clustering which has following steps-

1. Input: An event log, number of clusters, target fitness.
2. Getting all log data.
3. Define union of most frequent logs and set of log traces according to fitness value.
4. Create cluster set.
5. Add distinct process instances into the current cluster of a cluster set for first range fitness values of traces.
6. Looking ahead for remaining distinct process instances for that range.
7. Add remaining instances into current cluster.
8. Add new cluster into the cluster set for another fitness range values of traces.
9. Repeat steps 6 and 7.
10. Formation of number of clusters.
11. Output: A collection of event logs data represented by clusters of log.
12. Distance calculation method as follows :-

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Distance calculation method

V. RESULTS AND DISCUSSION

It shifts the measure of work per active trace based bug processing, where work is the quantity of whole number report performed as per users report details. At the point when tuples have succession numbers, taking care of copy accentuations is really simpler. The system allots the same grouping number to all copies of an event log data and bug report from previous record or dataset. At the point when an accentuation is a contender for submitting downstream, the system analyses the bug flow. In the event that the arrangement number is not exactly or equivalent to the last-submitted grouping number, then the accentuation is a copy and the merger drops it. This conduct is the same with respect to data checkpoints. This system processed the extracted textual data, and obtained the term-to-document matrix using parsing, filtering and term weighting methods.

Fig.5.1: List of all bugs cached from log data

Fig.5.2: Query output with fitness value

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

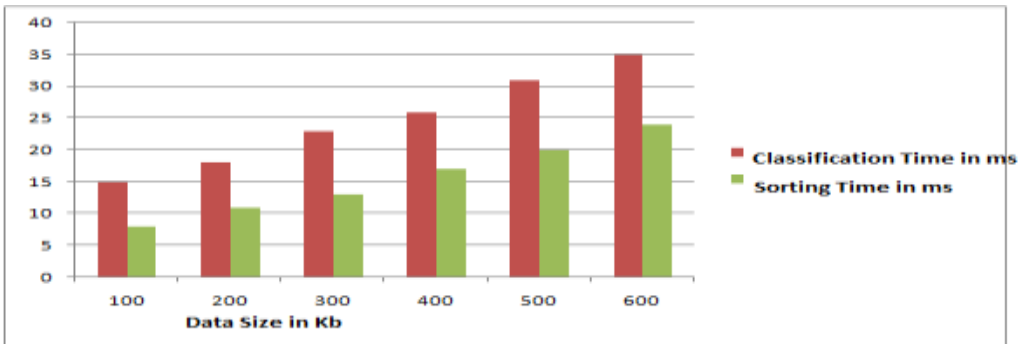

Fig.5.3: Time consumption to process data

VI. CONCLUSION AND FUTURE SCOPE

This system empirically investigates the data reduction for log data and bug data in bug repositories in a highly flexible real time environment. This proposed system improves system behavior functionality in flexible environments. This system extracts attributes of each log data set and bugs data set and train a predictive model based on real time data sets of a process. This system work provides an approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance. Active Tracing is an expensive step of software maintenance in both labor cost and time cost. To determine the order of applying process instance selection and feature selection for a new bug data set, this system extracts attributes of each log data set and bugs data set and train a predictive model based on historical data sets.

In future work, this system anticipate enhancing the consequences of information decrease in bug processing to investigate how to set up a high quality active trace information set and handle a space particular programming undertaking.

VII. REFERENCES

[1] Jochen De Weerdt, Seppe vanden Broucke, Jan Vanthienen, and Bart Baesens, “Active Trace Clustering for Improved Process Discovery”, IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 12, December 2013.
[2] Joachim Herbst, “An Inductive Approach to the Acquisition and Adaptation of Workflow Models” (1999).
[3] B. F. van Dongen and W. M. P. van der Aalst, Instance graphs, “Multi-phase Process mining: Aggregating Instance Graphs into EPCs and Petri Nets” (2005).
[4] Rakesh Agrawal , Johannes Gehrke , Dimitrios Gunopulos and Prabhakar Raghavan, Hierarchical clustering : “Automatic Subspace Clustering of High Dimensional Data” (2005).
[5] K. A. de Medeiros, A. J. M. M. Weijters, and W. M. P. van der Aalst, Genetic algorithms, “Genetic process mining: An experimental evaluation” (2007).
[6] Ferreira et al, Sequence Clustering,“Techniques for Process Mining Sequence clustering” (2007).
[7] Goedertier et al, Negative events, “Declarative Techniques for Modeling and Mining Business Processes” (2008).
[8] Christian W. Günther , “Activity Mining by Global Trace Segmentation” (2009).
[9] R.P. Jagadeesh Chandra Bose (JC) : “Abstractions in Process Mining: A Taxonomy of Patterns” (2009).
[10] Philip Weber, Behzad Bordbar, and Peter Ti˜no,“A Framework for the Analysis of Process Mining Algorithms” (2013)
[11] Jianmin Wang, Raymond K. Wong, Jianwei Ding, Qinlong Guo, and Lijie Wen, “Efficient Selection of Process Mining Algorithms", IEEE Trans. on Services Computing, ,vol.6 , pp. 484- 496, Dec. 2013.
[12] Marc Sole and Josep Carmona, “Region-Based Foldings in Process Discovery", IEEE Trans. on Knowledge and Data Eng., vol. 25, no. 1,pp. 192-205, Jan. 2013.
[13] Wil van der Aalst, “Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior", IEEE Trans. on Services Computing, November 2013.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

[14] Gianluigi Greco, Antonella Guzzo, Luigi Pontieri, and Domenico Sacca, "Discovering Expressive Process Models by Clustering Log Traces", IEEE Trans. on Knowledge and Data Engineering , vol 18, Issue 8, Page 1010-1027 , 2006.
[15] Stijn Goedertier, David Martens, Jan Vanthienen, Bart Baesens, "Robust Process Discovery with Artificial Negative Events", J. Machine Learning Research, vol. 10, pp. 1305-1340, 2009.
[16] Igor V. Cadez, David Heckerman, Christopher Meek, "Model-Based Clustering and Visualization of Navigation Patterns on a Web Site", Data Mining and Knowledge Discovery, vol. 7, no. 4, pp. 399-424, 2003.
[17] A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros, "Process Mining with the Heuristics Miner Algorithm", TU Eindhoven, BETA Working Paper Series 166, 2006.
[18] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster, "Workflow Mining: Discovering process models from event logs", IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9,pp. 1128-1142, Sept. 2004.
[19] M. Song and W.M.P. van der Aalst, "Towards Comprehensive Support for Organizational Mining", Decision Support Systems, vol. 46, no. 1, pp. 300-317, 2008.
[20] Wil Van Der Aalst, "Process Mining: Overview and Opportunities", ACM Transactions on Management Information Systems, Vol. 99, No. 99, Article 99, Feb. 2012.
[21] S. Goedertier, J. De Weerdt, D. Martens, J. Vanthienen, and B. Baesens, "Process Discovery in Event Logs: An Application in the Telecom Industry", Applied Soft Computing, vol. 11, no. 2, pp. 1697-1710, 2011.
[22] R.P. Jagadeesh Chandra Bose and W.M.P. van der Aalst, "Context Aware Trace Clustering: Towards Improving Process Mining Results", Proc. SIAM Int'l Conf. Data Mining (SDM), pp. 401-412, 2009.
[23] G.M. Veiga and D.R. Ferreira, "Understanding Spaghetti Models with Sequence Clustering for Prom", Proc. Int'l Business Process Management Workshops, pp. 92- 103, 2010.
[24] R.P. Jagadeesh Chandra Bose and W.M.P. van der Aalst, "Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models", Proc. Int'l Business Process Management Workshops, pp. 170-181, 2009.
[25] T. Hofmann and J.M. Buhmann, "Active Data Clustering", Proc. Neural Information Processing Systems Conf. (NIPS), 1997.

CITE AN ARTICLE

Sonawane , Swapnali , and D. S. Kulkarni, Prof. "CONSTRUCTION OF IMPROVED PROCESS MODELS BY CLUSTERING EVENT LOGS." *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY* 6.7 (2017): 516-21. Web. 15 July 2017.