Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.11, No.4, 2020
www.iiste.org

Generation of Process Sequence Based on Implicit Temporal Overlap Function

Linda Uchenna Oghenekaro* Chidiebere Ugwu Laeticia Nneka Onyejegbu
Department of Computer Science, University of Port Harcourt, PMB5323, Rivers, Nigeria

Abstract:
Activities within processes occur in sequence, and the discovering of these sequences is an essential step and of great significance to process mining. This paper is aimed at intelligently discovering process sequences that lie within the helpdesk unit event log, which was primarily obtained from the 4TU repository. Explicit approaches have mostly being applied to mining rules and little attention given to sequences that can be generated via implicit approach. Hence, an implicit approach to association rule discovery was adopted using the modified temporal overlap scoring module (TOSM). The module was implemented using Java programming language. The experimental results showed that the temporal overlap module discovered sequences in an intelligent manner by factoring in the overlap property and identifying hidden dependencies. The resulting association rule generated for each sequence, as represented in the lift value, was recorded as significant to the entire log as compared to that of the explicit approach.
Keywords: Hierarchical Temporal Memory, Overlap, Process Aware Information System, Event Logs, Process Sequence.
DOI: 10.7176/CEIS/11-4-04
Publication date: June 30th 2020

1. Introduction

Help Desk unit which is also referred to as the end-user single point of contact (SPOC), is a vital resource and a solution for all businesses, because it assists customers in all suitable ways, handling both queries to complaints. As the saying goes, Customer is King; hence, having satisfied customers, is a clear indication for a company's long-term profitability and on the other hand, epileptic processes in the help desk unit of every company can lead to customer frustration and in worse cases, the customer can be lost forever (Dumas et al., 2013). The principal aim of the help desk unit is to reduce business function's downtime significantly and keep customers informed on the services of the organization. The ability to respond to customers in a timely manner helps them feel valued and improve their customer experience with the company. However, help desk processes at large scale businesses are very expensive, with regards to labor hours spent and cost of hourly pay (Aleem et al., 2015). The workload of help desk unit is very dynamic, employing a sufficiently large team of helpdesk staff can lead to several staff being idle during help desk operations and on the other hand, employing few staff will result in delay of issue resolution. Hence, overview of helpdesk processes helps plan staffing levels and minimized labor cost. Businesses are more dependent on IT and complexity of technology applied is increasing at a fast rate, all these developments have led to customers demanding more value for their money, hence the need to intelligently monitor the process of the unit that has direct impact on the customer. The business environment has grown to be a very competitive one, with every business trying to guard its existing customers and at same time grow its customer base. This can only be achieved with an intelligent and functioning help desk unit. The ability to accept huge amounts of query data and process or resolve issues within a short time interval is a measure of the quality of a help desk. The helpdesk however, needs to be monitored so as to improve the quality of the processes that lie therein.

Event logs refer to data about business processes occurring during the system's performance and they are collected and stored in the information systems which are advantageously used as input information for building and retrieving business process model (Grigorovia et. al., 2017). In event logs, each event refers to a case, an activity and a point in time. They are also said to be data that reveals the real events that has taken place rather than how it is supposed to be or how it is perceived by its authors. Event logs come from a wide variety of sources such as; patient data in a hospital, financial data on spreadsheets, transaction logs from trading systems, message log from middleware and a host of others (Dogen, 2011). Although event logs are available in information systems of organizations, they frequently lack the understanding of their real-life processes.

The knowledge hidden within these event logs can be converted into useful information (van der Aalst, 2012). Figure 1 visualizes these definitions. An event log consists of events, which contain activities that can be seen in the IT systems and mapped to the process model. Every event is mapped to a case, a specific execution of the process. Every case has a sequence of ordered events (Dogen, 2011). Each sequence forms a variant. Different people in different cases execute these activities in a different order (Elzinga et al., 1995). Beside case identifiers, event names and timestamps, an event log can contain additional event properties, such as costs and resources (participants of the process), internet protocol (IP) addresses and much more (Kalenkova et al., 2017). To discover a control flow, an event log is represented as a multiset of traces, each of which is a sequence of events,

corresponding to a concrete case identifier:

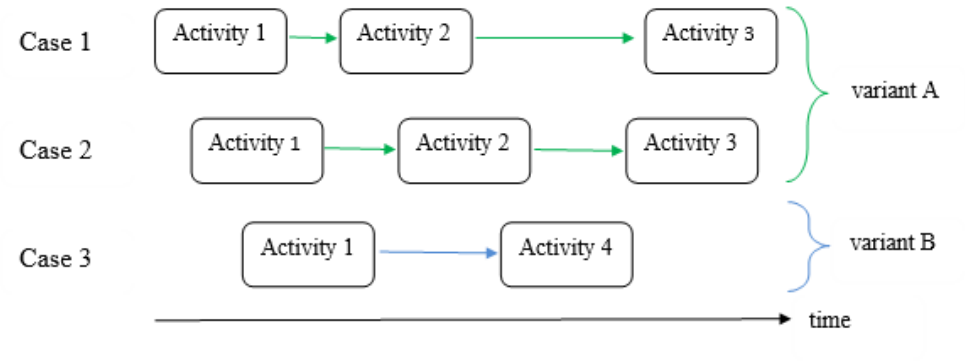

Figure 1: Visualization of CaseID, Activity and Variant (van der Aalst, 2012)

2. Related Literature

Customer service is one key factor in building a company's reputation and brand. This service is offered by the help desk unit, which is a primary unit of every organization, aimed at resolving customer's queries and complaints in an organized and systematic way to ensure customer satisfaction. Efficient functioning of the help desk unit aids organization in enhancing service quality. Figure 2 illustrate the various channels through which customers can log their complaints. Once the communication channels are identified, the next step is to funnel them down to create cases (Verenich, 2016). The figure shows customers request coming from various channels such as online inquiry, call support, email messaging and direct call in, each of this form is eventually classified as a case.

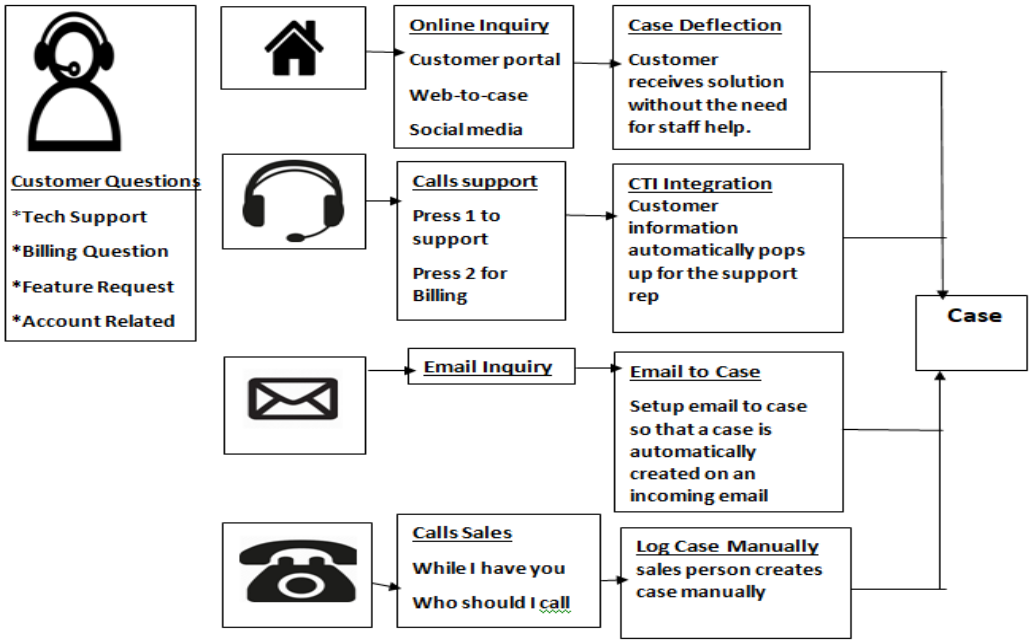

Figure 2: Customer Communication Channels (Verenich, 2016)

Improvement of customer digital experience has been sort by various researchers, Osman & Ghiran (2019), proposed a novel approach to generate customer traces from information systems such as customer relationship management systems (CRMS) in a bid to improve customer experience. Their work highlighted that IT methods, tools and methodologies are needed to better understand customer's needs. Hinshaw (2012), in his article, highlighted some tools and methodologies that helps to improve customer experience, such of these include; design thinking, which is a methodology used by software designers in high-tech companies to assist them in better understanding the needs of the customers so as to provide desirable solutions. Other tools such as Big Data Analytics, Business Process Management, Machine Learning, Internet of Things and several others are being used optimally. Design of Customer Journey Maps (CJM) have posed a big challenge for companies as it involves a lot of integration such as web and mobile, to allow for interaction between company and client (Osman & Ghiran, 2019). SAP and Oxford Economics carried out a research in 2017 which involved over 3100 companies from all

over the world and these companies cut across all domains such as manufacturing, health, and banking. Their study showed that 96% of these companies have increased investment on technologies that helps to improve customer's digital experience.

Literature highlights an abundant of approaches of data mining techniques applied on customer relationship management systems (CRMS) including classification, association, regression, clustering, visualization, predictive analysis or sequence discovery. All these approaches were relevant to study; however, none of them provides a process-centric approach. An agile methodology known as Design Thinking (DT) has become very popular in the last decade because of its cognitive power to easily adapt to change and give visual overview to company's processes (Osman & Ghiran, 2019). Sakchaikun et al. (2018) applied three process discovery techniques to real life help desk event log previously extracted from the help desk unit of on IT company. These techniques include social network miner, time performance fuzzy miner supported by disco fluxicon tool and RapidProm tool. The helpdesk unit, to which these techniques were applied, was manned by 5 helpdesk staff. The retrieved data, which include the event processes and the staff id of the resource persons, were pre-processed, filtered and subjected to the process mining tools. Results showed the average SLA time recorded between the opening of a ticket and the close of the ticket was 4 days against the expected 4 hours according to the guideline of the company. Further findings revealed that of the five helpdesk staff, only four really handled the workload of the department; the last staff performed only 5 task in the entire year, probing the behavior of this last staff revealed that he used admin privilege to reassign all his task to the other four staff, hence, playing absolutely an inactive and idle role all year round (Sakchaikun et al., 2018). Results of the study assisted the company in improving their customer service efficiency. The life cycle of process management is characterized by an iterative set of activities. The cycle is repeated for various process instances. Before drawing up a process lifecycle, the organization's goals must be fully understood and current processes must be identified.

3. Process Aware Information System

The PAIS can be defined as a system driven by process model (Rovani, et. al., 2005). It serves as a bridge that connects people and software through process technology. There are various software that can be used to create PAIS which include: WorkFlow Management (WFM), Customer Relation Management (CRM), Supply Chain Management (SCM), Product Data Management (PDM) and Enterprise Resource Planning (ERP) (van der Aalst, et. al, 2012). They have a built-in workflow component that is responsible for generating event logs. WFMS was birthed in the 90s and focused on proffering ways to automate tasks integrated to a human activity and to control the flow of information. In following years, BPMS emerged and became an extension of WFMS, however, it focused mainly on management roles, operation analysis and organization work. However, the applications of WFMS or BPMS are very limited in many organizations owing to the difficulties in dealing with semi-structured or unstructured processes.

3.1 Event Logs

Events are produced at transition firings, the sequence of events produced by the system is exactly the sequence of transition firings (Cook et al., 2004). On process-aware information systems (PAISs), these events have log file known as event logs. Event log is the start point of business process model analysis. When carrying out process mining for the reason of process discovery, it means there exist no previous model and hence, the event logs serves as the basic resource that helps to provide information about business process activities (Sarno and Effendi, 2017). Event logs can be gotten from large-scale information systems such as Customer Relationship Management (CRM), Enterprise Resource Planning (ERP) and Workflow Management Systems (WFMS). During the execution of business process, information of each activity and abstract procedure are recorded into these information systems. Event logs contain several information depending on the organizational information (Sarno et al., 2016). In general, event log is divided into three basic parts; Case identification number, the timestamp and the activity as illustrated in table 1 (Sutrisnowati et al., 2014).

I) Case: this refers to a record of events related to a single executed process instance. It can also be described as the number of times an item is being executed.
II) Timestamp: This, records a time of events that belong to same case.
III) Activity: this is a part of an event log that represents the production process of a product

Table 1 also include other attributes such as resource and cost, this shows that there are several other attributes that exist within the log, however, the basic attributes used in mining are case ID, timestamp and activity. The table shows a resource person identified as Emeka, carry out an activity A at 11.10, same resource person carried out all the activities within the segment of this table, using the timestamp and event id a graphical representation of the execution of these processes can be generated.

Table 1: A fragment of an event log (Sutrisnowati et al., 2014)

Case ID	Properties				
	Timestamp	Activity	Resource	…	…
35654423	20-10-2018:11.10	A	Emeka	…	…
35654424	20-10-2018:13.21	B	Emeka	…	…
35654425	20-10-2018:13.32	C	Emeka	…	…
35654426	20-10-2018:15.05	D	Emeka	…	…
35654427	20-10-2018:15.55	E	Emeka	…	…
…	…	…	…	…	…

4. Materials and Method

The materials and method used in this paper was described in this section. Their individual and collective contribution to the work was also highlighted.

4.1 Dataset

The dataset was primarily obtained from synthetic event logs of the process mining 4TU Data centre, which was same data used by Ayo, et al. The dataset consists of helpdesk processes that have eight different activities being carried out at several different timestamps. The dataset comes from the Process Aware Information System (PAIS), and it is saved on the system in a comma separated value (CSV) format. Within the dataset, the different activities have been coded using short names as thus: a = register, b = examine casually, c = examine thoroughly, d = check ticket, e = decide, f = reinitiate request, g = pay compensation and h = archive request. Every case begins with the registration of the customer's request and is properly ended with the pay compensation activity or archive request activity. There are six attributes on the dataset, which include; case ID, event ID, timestamp, activity, resource and cost. Every row in the dataset represents an event which is identified using an event ID. Related rows are being connected using the event ID and the case ID.

4.2 Temporal Overlap

The HTM theory is an online machine learning concept developed by Hawkins J. and George D. of Numenta in 2004. The model copies some of the algorithmic and structural properties of the neocortex, which is the seat of intelligence in the human brain. In the book *On Intelligence*, (Hawkins & Blakeslee, 2004), Hawkins developed the idea to build a simple model of the neocortex not by simulating every part of it but by reducing it to its core function (Galetzka, 2014). The temporal aspect of the theory accounts for the activities that occur in a given order and it considers how the present meaning of the activities affects the next activity. The overlap scoring, accounts for associated activities with significant permanencies. The entire module was executed in java programming language and it handled the generation of the process sequences by executing an algorithm. The TOSM algorithm was adopted to improve the process sequence generation of an existing system developed by Ayo et al., 2017, where the Bayesian Scoring Function (BSF) was employed in explicitly defining the rules, however, it had a shortcoming of having its association rules for the process sequence being defined by the system developer, which implied that the robust nature of their system depended on the definition of the rules. This disadvantage poses a huge risk to real life processes which takes its toll in any direction. The existing algorithm is displayed in algorithm 1 and the generated values for its rules are displayed on table 2. The TOSM module started with initializing variables, followed by the generation of process sequences by implementing the TOSMEventLog algorithm, as seen in algorithm 2. The algorithm was modified from an overlap scoring metric proposed earlier in 2015 (Ahmad & Hawkins). The modification introduced a random perturbation factor r_b, to account for the entropy that may arise during the formation of overlapping patterns by the HTM. The CSV file was introduced into the execution of the TOSM using a java import statement. The corresponding sequences where then generated using the TOSMEventLog algorithm and its algorithm keeps account of its temporal variability.

Algorithm 1: BSFEventLog.
Input : Event Log
Output : CompleteLog
Process :
1. Input Event log
2. Definition of Association Rule N
3. for i = CasesAtLog length
4. for j = ActivitiesAtLog length
5. for each association rule v
6. Compute Scoring function
7. if Deg Of Conf > = 0.5
8. for k = 1 to N

9. if (predecessor (v) exists without successor(v))
10. insert successor(v) at i
11. Return CompleteEventLog

Table 2: Association rules and their respective scoring functions (Ayo, et al., 2017)

Association Rules	Degree of Support	Degree of Confidence	Lift Value
a -> b	1.00	1.00	1.00
a -> b -> c	1.00	1.00	1.00
a -> b -> c -> d	0.47	0.47	0.73
a -> b -> c -> d -> e	0.22	0.45	0.45
a -> b -> c -> d -> e -> f	0.20	0.93	0.93
a -> b -> c -> d -> e -> f -> g	0.18	0.94	0.94
a -> b -> c -> d -> e -> f -> h	0.01	0.06	0.22
a -> b -> c -> d -> e -> h -> g	0.01	1.00	1.00
a -> b -> c -> d -> e -> h	0.02	0.06	0.24
a -> c -> d -> f -> e -> h -> f	0.01	1.00	1.00

Algorithm 2: TOSMEventLog.
Input: Event Log
Output: CompleteEventLog
Process:

1. Input Event log, Define breaking overlap threshold, o_b and a random factor, r_b
2. Number of Process Sequences N
3. for i = CasesAtLog length
4. for j =ActivitiesAtLog length
5. Compute Causality Process Sequence of the form A, A->B, AB->C, ABC -> D
6. for each process sequence, *s*
7. Compute an Overlap Scoring function, o_{th} based on *s*
8. if $o_{th} >= o_b + r_b$
9. for k = 1 to N
10. if (predecessor (v) exists without successor(v))
11 insert successor(v) at I // the successor is predicted
12. Return CompleteEventLog

The values used to determine how relevant an association rule is to the universal set in terms of its degree of support and confidence was also evaluated using their mathematical notation as seen in equation 1 and 2. The lift value was also evaluated as the ratio of the confidence value to the support.

$$Supp\ (A, B) = S\ (A\ \cup\ B\ ) = \frac{x}{n};\ A \cap B = \varphi \qquad (1)$$

$$Conf\ (A, B\ ) = \frac{S\ (A\ ,B)}{S\ (A\ )} \qquad (2)$$

5. Result and Discussion

After the execution of the TOSM block on the CSV file, a set of association rules was generated and each association rule was measured against degree of confidence, support and lift. These degrees served as the numerical attributes to each rule and their values range between 0 and 1, the more significant the rule is the higher the value and vice versa. Given that not every rule is significant to the entire set. The degree of Support, denoted as Supp(A,B), gives the percentage of traces containing all items present in the association rule. The mathematical notation is given in equation 1. While the degree of Confidence, denoted as Conf(A,B), depicts the frequency of B in all instances that A is present, where A and B are activities in the event log, it is mathematically demonstrated in equation 2. Lift, which is the resultant attribute of Confidence and Support, is the fraction of the rule confidence to B support. The output of the TOSM was dropped on the system's drive. Table 3 shows the output file, which consists of the process sequences and their corresponding function scores. The fourth association rule, a -> c -> d -> e -> h, on one hand, can be seen to be very low on the degree of Confidence with a score of 0.0531, which shows that the confidence of this rule is low and implied that, there might hardly be a second trace of this form within the event logs. The first rule, a -> b, on the other hand, shows high degree of support and confidence, which eventually leads to a high lift value. This numbers implies that the rule is significant to the event log. The rules generated with the TOSM are more than those of the BSF, where rules were explicitly defined as seen in table 2 and table 3; this implies that there exist other rules that were not defined using the Bayesian scoring approach. Hence, table 4 which holds the summary of findings shows the total number of rules discovered from both approaches and the average the degree of the association as calculated from the lift value between 0.6 and 1.0. The

TOSM recorded 7 more rule discoveries than the BSF, and an average degree that was higher than that of the explicit approach by 0.06.

Table 3: Association rules and their respective scoring functions

Association Rules	Degree of Support	Degree of Confidence	Lift Value
a -> b	0.8113	0.7003	0.8631
a -> b -> c	0.7456	0.0345	0.4627
a -> b -> d -> h	0.4781	0.4781	1.0003
a -> c -> d -> e -> h	0.2156	0.0531	0.2462
a -> b -> d -> e -> g	0.9784	0.5542	0.6564
a -> c -> d -> e -> h	0.4788	0.4784	0.9916
a -> c -> d -> f -> h	0.8152	0.4509	0.5531
a -> b -> d -> e -> g -> h	0.9831	0.9388	0.9549
a -> b -> c -> d -> b -> e -> g	0.9884	0.9412	0.9522
a -> c -> d -> f -> e -> f -> g	0.7125	0.0653	0.9162
a -> c -> d -> f -> e -> f -> g -> h -> g	0.7456	0.0654	0.8771
a -> b -> d -> c -> e -> f -> g -> h -> g	0.8147	0.6521	0.8002
a -> b -> d -> f -> e -> h -> g -> h -> g	0.6125	0.0467	0.7624
a -> c -> d -> c -> b -> d -> e -> f	0.6337	0.0677	0.6833
a -> d -> b -> d -> e -> f -> g -> h -> g	0.7193	0.4996	0.6945
a -> b -> c -> d -> e -> f -> e -> g -> h	0.7456	0.0657	0.8811
a -> c -> b -> c -> d -> e -> f -> g -> h	0.2146	0.1211	0.5643

Table 4: Summary of Findings

Rule Discovery Approach	Number of Rules Discovered	Average Degree of High Association
Implicit (TOSM)	17	0.88
Explicit (BSF)	10	0.82

6. Conclusion

The help desk unit is an important aspect of every organization that intends to carry out business processes in an intelligent and proactive manner. The temporal memory component of the hierarchical temporal memory had been modified to address the peculiarities of the event logs in the helpdesk unit of organizations. The TOSM leverages on the sparse nature of the hierarchical regions to extract process sequences that are significant to the entire log. From which robust process model can subsequently be produced. The modified algorithm was implemented on helpdesk event logs extracted from the PAIS, the helpdesk unit was selected as a result of its crucial role it plays in every organization. The long term sustainability of every institution depends on the disposition of its end users. The modified algorithm will boost subsequent activities employed in mining processes, given that low and irrelevant associated process sequences will not be selected.

References

Aleem, S., L.F. Capretz, F. Ahmed (2015). Business process mining approaches: a relative comparison. *International Journalof Science, Technology and Management.*1557-1564.

Ayo, F.E., O. Folorunso, F. T. Ibharalu (2018) A probabilistic approach to event log completeness. *Expert Systems With Applications.* 80(1), pp. 263–272.

Cook, J.E., Z. Du, C. Liu, A. L. Wolf (2004). Discovering models of behavior for concurrent workflows. *Computer Industry, Elsevier.*53(7):297-319.

Dongen, B.F. (2011). Real-life event logs - Hospital log. *Eindhoven University of Technology Dataset. doi:10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5*

Dumas M., M. La Rosa, J. Mendling, H. A. Reijers (2013). Fundamentals of Business Process Management. 1st edn. Springer, Heidelberg.

Elzinga, D.J., T. Horak, C.Y Lee,C. Bruner(1995). Business process management: survey and methodology. *IEEE Transaction EngineeringManagement.*42(2).119–128.

Fahland, D., W.M.P. van der Aalst (2012). Simplifying discovered process models in a controlled manner. Information Systems 38(5), 585-605.

Galetzka, M. (2014). *Intelligent Prediction: An empirical study of the Cortical Learning Algorithm.* Master Thesis, University of Applied Sciences Mannheim.

Grigorova, K., E. Malysheva, S. Bobrovskiy (2017). Application of Data Mining and Process Mining approaches for improving e-Learning Processes. *In: 3rd International Conference "Information Technology and Nanotechnology".*

Hawkins, J. and Blakeslee, S. (2004). *On Intelligence.* Henry Holt and Company.
Kalenkova, A. A., W.M.P. van der Aalst, I.A. Lomazova, V.A Rubin (2017). Process mining using BPMN: relating event logs and process models. *Software and Systems Modeling.*
Rovani, M., F.M Maggi, M. Leoni, W.M.P van der Aalst (2015). Declarative process mining in healthcare. *Expert Systems Applications.* 42(5): 9236-9251.
Sarno, R. and Y.A. Effendi (2017). Hierarchy Process Mining from Multi-source Logs. *Telkomnika*. 15(4). 1960-1975.
Sutrisnowati, R. A., H. Bae, L. Dongha, K. Minsoo(2014). Process Model Discovery based on ActivityLifespan. *International Conference on Technology Innovation and Industrial Management Seoul*. 137-156.
van der Aalst, W.M.P. (2012). Process mining: *Communication of the Association for Computing Machinery.* 55(8): 76-83.
Verenich, I (2016), “Helpdesk”, Mendeley Data, v1https://dx.doi.org/10.17632/39bp3vv62t.1