



GAE: A Novel Approach for Software Workflow Improvement by Unhiding Hidden Transactions

Shashank Sharma^(✉) and Sumit Srivastava

Manipal University Jaipur, Jaipur, India
shashanksharmaaa@gmail.com

1 Introduction

Workflow mining a.k.a. process mining is a considerably new and emerging area of academic research within data analytics. The key objective here is to deploy workflow-related data in the direction to obtain pertinent info and knowledge by employing data analytic algorithms and determining a workflow model. This section discusses the concept of how an event log is the foundation of exploration along with other main building blocks of process mining. Work process mining articles to course the hole between huge information examination and traditional business work process/process administration. This field can fundamentally be categorized into (1) workflow revelation, (2) conformance checking, and (3) upgrade [1]. This permits the extraction of experiences about the by and large and internal conduct contained in any given procedure. Work process disclosure strategies underline on utilizing the occasion information in order to decide work process models. Conformance checking procedures underline on supporting the occasion information on a work process model to confirm how well the model fits the information and the other way around [2]. Despite the fact that increase strategies utilize occasion information and work process models to repair or expand the work process show. Thus, work process mining gives the course the crevice between information mining and machine learning hones and the business procedure administration teach.

1.1 The Event Log as Main Focus of Analysis

Data is a crucial building block in various discovery domains. Process Mining uses data that is accounted by event logs. A sample log is shown in Table 1. An event log in terms of this situation can be defined as the process of recording of an action instance on the system. Action or activity instances are units of work that are registered by the system when work is piloted in the situation of an assured process. Statuses of activity or action are specified to a set of languages that are fixed to the workflow modeling hypothesis. Various workflow modeling hypothesis have unique implementation standards that are governed by their state evolution diagrams. A simple state evolution diagram has been explained in Fig. 1. More elaborate diagrams have been described in process modeling literature. Consider an example that the Object Management Group

(OMG) characterizes the life cycle of an action in its Business Process Modeling Notation (BPMN) particular [3]. Adding to this, different process displaying ideal models, for example, YAWL (Yet Another Workflow Language), case taking care of [4] and revelatory methodologies (EM—BrA2CE [5], Declare [6]) which proposes positive however similar to base representations for their relating delineation semantics.

Table 1. An example of typical event log

Case ID	Activity name	Event type	Originator	Timestamp	Extra data
011	Make order form	Start	Employee A1	10-19-1955 15:15:02
012	Make order form	Complete	Employee A2	10-19-1955 05:14:01
....
....

It can get fascinating to experience different establishment semantics and state change charts, for work process mining and work process examination; notwithstanding, the real information is what is significant. Process data is generally pooled from various vaults that are gotten from CRM, ERP, WFM, and other different data frameworks. This outcome is in trouble while recognizing hypothetical establishment of an occasion as far as state progress charts and genuine information found practically speaking. In various sectors such as CRM (customer relationship management), product development, financial services, etc., business workflow depends heavily on legacy information systems or low workflow-oriented information systems. Further, the registered business process data is well defined such that only a certain type of state transition (e.g. accomplishment of an activity occurrence) can be mined via. actual data blocks. Workflow mining can be considered most fruitful in modular conditions where business information systems have broader options of behavior, here; the transition of accessible data into an event log is most often a significant task.

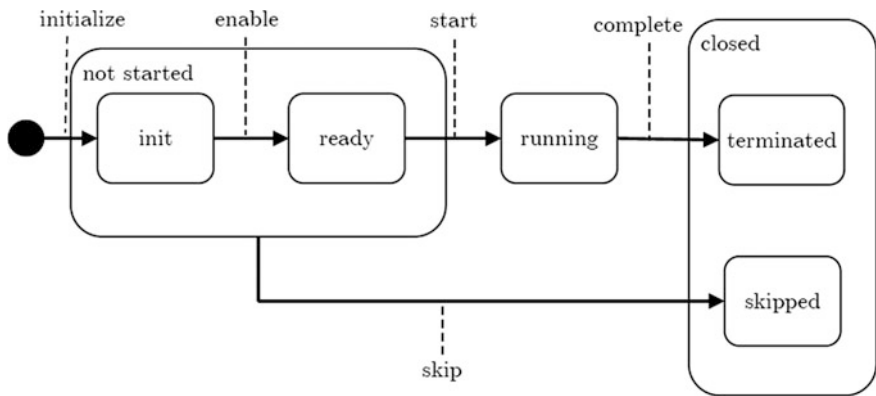


Fig. 1. A basic state transition diagram

Filtration and extraction of event logs through business information systems is usually carried out by text-based data scripts. In reality, process data is more often distributed over various data sources and it is painstaking to define the precise scope of process that is studied. Further, a comprehensive ETL-phase is required before a concrete analysis is initiated. Adding to which, the data must be in an event log storage format.

One of the underlying ways to deal with putting away log in view of occasion performed/executed is in the MXML design (Mining Extensible Markup Language). Since 2003, the MXML organize has been utilized as the reality-based (de facto) standard since it is exceedingly inter-ground with the ProM-structure which is a system utilized for scholarly reason for work process change. It was not up to this point IEEE team was favored rather than MXML for work process mining. This new configuration, XES (eXtensible Event Stream) enhances the first standard, since it is less prohibitive. The meta-model of XES is portrayed in [7].

As occasion logs are the establishment of the approach of workflow mining, it is significant to express the variable necessities to which an occasion log must approve. After taking these three suppositions are required and basic:

- Activity instance must be well defined for the workflow instance of an event which specified by a unique activity name
- Unique process instances or ID which are used by cases must be referred by an events
- Ordered Timestamp must be recorded by the events.

2 Literature Review

As Agrawal et al. [8] and Pinter and Golani [9] are the fundamental ones that don't explicitly get the possibility of the split/join centers in the mined models. The reason is that they concentrate on a model for the Flow check work structure [10] and each point in this system has an OR-part/join semantics. In fact, each organized round segment in the model has a Boolean limit that surveys to real or false after an errand is executed. The appraisal of the Boolean conditions sets what number of branches are established after an endeavor is executed. Cook et al. [11] have the primary approach that does not concentrate on a whole mined model. Their approach looks for the most progressive cases in the model. In actuality, all over they do mine a whole procedure appear, however that is not their essential point. The build ups that can't be mined by all frameworks are circles, sans non-choice, imperceptible assignments moreover, duplicate endeavors. Grecco et al. [12] can't mine any kind of circles. The reason is that they show that the models their computations delve think about as small extra lead (that is not in the event log) as could be normal the situation being what it is. They do as such by posting each one of the takes after that the mined model can deliver and differentiating them and the follows in the event log. Models with circles would make this endeavor outlandish. Some different procedures can't mine self-assertive loops in light of the fact that their model documentation (or portrayal) does not bolster this sort of

loops. The fundamental motivation behind why most strategies can't mine non-neighborhood sans non-decision is that the majority of their mining calculations depend on neighborhood data in the logs. The systems that don't mine nearby without non-decision can't do as such on the grounds that their portrayal does not bolster such a develop. Generally the system depends on a piece organized documentation, as Herbst et al. also, Schimm. Skip errands are not mined in light of depiction limitations as well. Split/join imperceptible endeavors are not mined by various techniques, except for Schimm and Herbst et al. Actually, we similarly don't center at finding such kind of assignments. Nevertheless, it is routinely the case that it is possible to make an exhibit with no split/join imperceptible errands that communicates a comparable direct as in the model with the split/join subtle assignments. Duplicate endeavors are certainly not mined in light of the way that various methods expect that the mapping between the endeavors what's more, their imprints is injective. By the day's end, the names are exceptional per task. The principal strategies that mine duplicate endeavors are Cook et al. [11] for progressive structures just, and Herbst et al. [13] for both back-to-back and parallel techniques. We don't consider Schimm [4] to mine technique models with duplicate errands since his approach expect that the revelation of the duplicate endeavors is done in a preplanning step. This movement identifies every one of the duplicates and guarantees that they have unique identifiers when the event log is given as commitment to the mining calculation. As a matter of fact, every procedure that we survey here would handle copy undertakings if this same pre-preparing step would be done before the sign in and is given as contribution to them.

3 Proposed Algorithm with Procedure with a Sample Case Study

With a specific end goal to apply a hereditary calculation, we have to speak to people. Every individual compares to a conceivable procedure model and its portrayal ought to be anything but difficult to deal with. Our underlying thought was to speak to forms straightforwardly by Petri nets. Tragically, Petri nets end up being a less helpful way to speak to forms in this specific circumstance. So to overcome this problem we use causal matrix [3, 14].

Table 2. Concise and concrete encoding format of an individual in Table 2

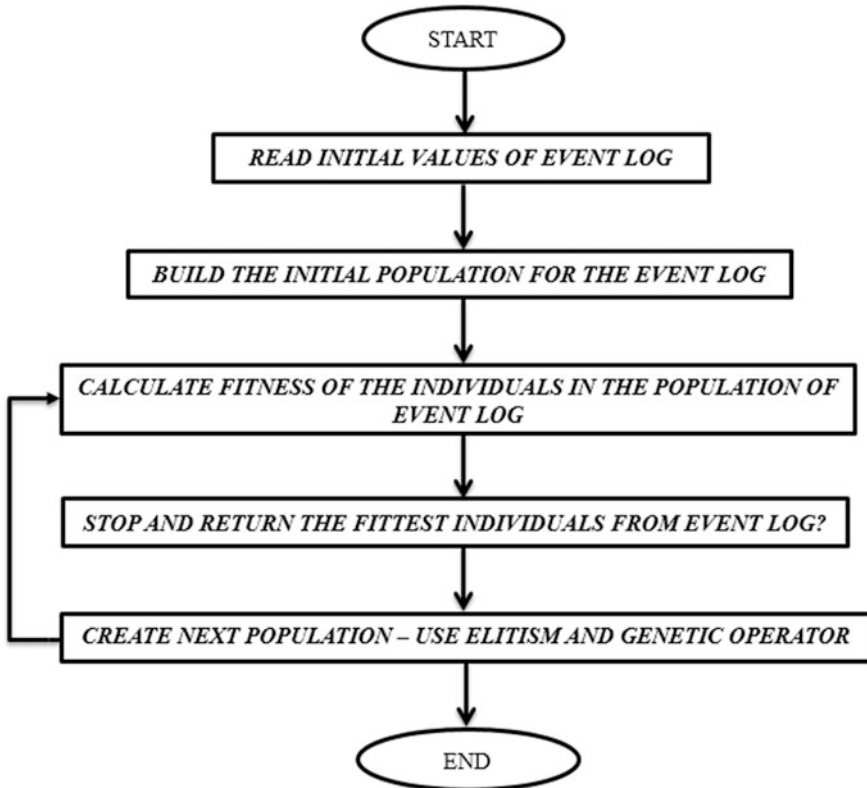
Activity	Input	Output
A	{}	{{B, C, D}}
B	{{A}}	{{H}}
C	{{A}}	{{H}}
D	{{A}}	{{E}}, {{F}}
E	{{D}}	{{G}}
F	{{D}}	{{G}}
G	{{E}}, {{F}}	{{H}}
H	{{B, C, G}}	{}

A matrix of causality is a row, X with elements named as (M, N, O, P) where

- M comprises of activities with type of finite sets,
- $N \subseteq M \times M$ is the relation of causality,
- $O \in A \rightarrow P(P(M))$ is the function of input condition type, 3
- $O \in A \rightarrow P(P(M))$ is the function of output condition type,

such that

- $N = \{(m1, m2) \in M \times M \mid m1 \in O(m2)\}$, 4
- $N = \{(m1, m2) \in M \times M \mid m2 \in P(m1)\}$,
- $\forall m \in M \forall Q, QR \in O(m) \quad Q \cap QL = \emptyset \Rightarrow Q = QL$,
- $\forall m \in M \forall Q, QR \in P(m) \quad Q \cap QL = \emptyset \Rightarrow Q = QL$,
- $N \cup \{(mo, mi) \in M \times M \mid mo \bullet N = \emptyset \wedge N \bullet mi = \emptyset\}$ is a connected graph of strong type. The proposed process efficient GAE (Genetic Algorithm for Events) is based on GA and process mining of event logs. In this, fitness function is computed



Flowchart 1: Proposed algorithm for GAE

by the quality of an individual. The quality of an individual is basically set by its replaying of the log traces. This semantics permits us additionally to characterize an idea of wellness required for the hereditary calculations. Details regarding the calculations of the various parameters are beyond the scope of this paper. Utilizing the Petri net portrayal, we can play the “token amusement” to perceive how every occasion follow in the log fits the individual spoke to by a causal network. The entire flow of GAE is shown in Flowchart 1. The steps of the proposed algorithm are as follows:

In this, first we take the event log of hospital case of healthcare information system. In this particular event log we have 42 events or events flow. Our aim is to classify the events on the basis of type of events. We start with event log then we select MXML legacy classifier and process discovery algorithm for the extraction of Petri net. Along with this, we apply some user-specified constraints to get expected result (Fig. 2, Table 3).

Table 3. Causal matrix made and utilized for portrayal

Input									Output
	true	A	A	A	D	D	$E \wedge F$	$B \vee C \vee C$	
\rightarrow	A	B	C	D	E	F	G	H	
A	0	1	1	1	0	0	0	0	$B \vee C \vee D$
B	0	0	0	0	0	0	0	1	H
C	0	0	0	0	0	0	0	1	H
D	0	0	0	0	1	1	0	0	$E \wedge F$
E	0	0	0	0	0	0	1	0	G
F	0	0	0	0	0	0	1	0	G
G	0	0	0	0	0	0	0	1	H
H	0	0	0	0	0	0	0	0	true

4 Comparison Analysis

For verifying and validating the effectiveness of GAE, we used the standard algorithm which are used for workflow management as well as analyze and compare it with proposed GAE. In figure number 5, you find black boxes (dark or complete black), which referred to hidden transactions, which shows that other or previous algorithms are unable to extract or locate hidden transactions although algorithms are able to classify the event log (Figs. 3, 4, 5 and 6, Table 4).

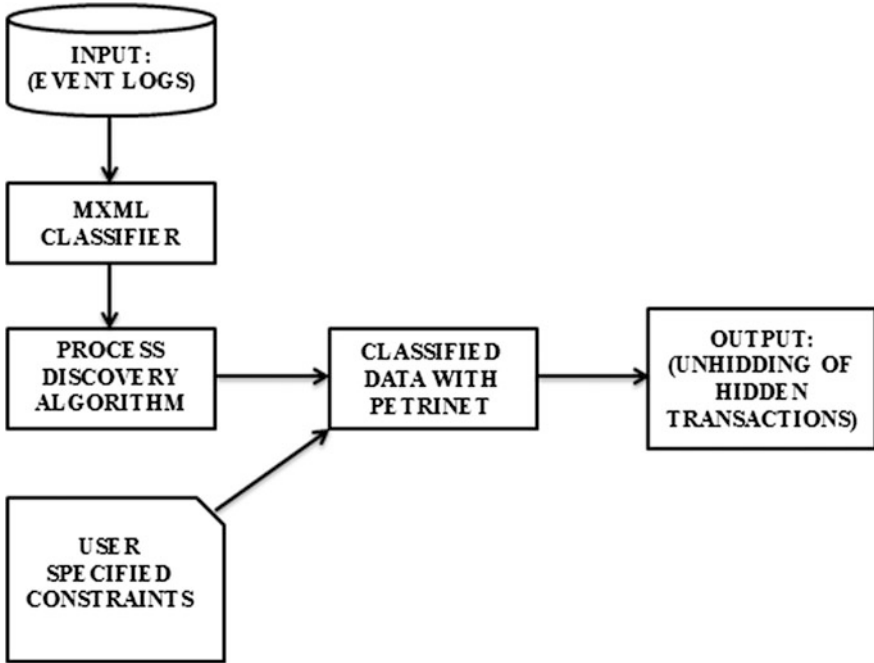


Fig. 2. Procedure for GAE

Table 4. Comparison of result for various algorithms

Name of algorithm	Input format	Output format	Intermediate output	Able to unhide hidden transactions
Alpha	MXML log file	Petri net	No	No
Alpha++	MXML log file	Petri net	No	No
Tsinghua-Alpha	MXML log file	Petri net	No	No
GAE	MXML log file	Petri net	Heuristic Net	Yes

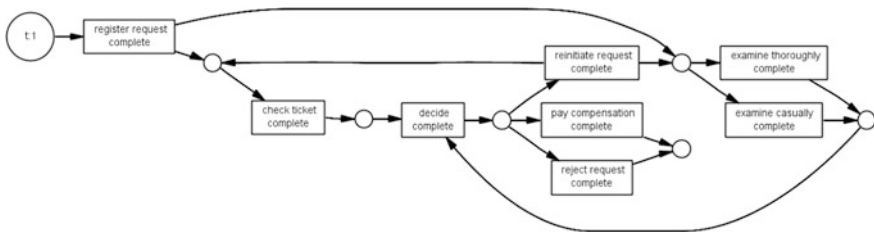


Fig. 3. Alpha algorithm

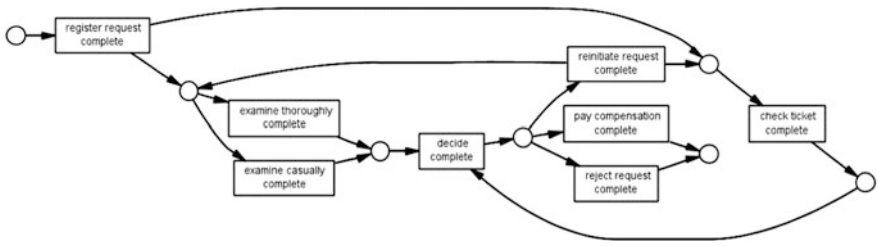


Fig. 4. Alpha++ algorithm

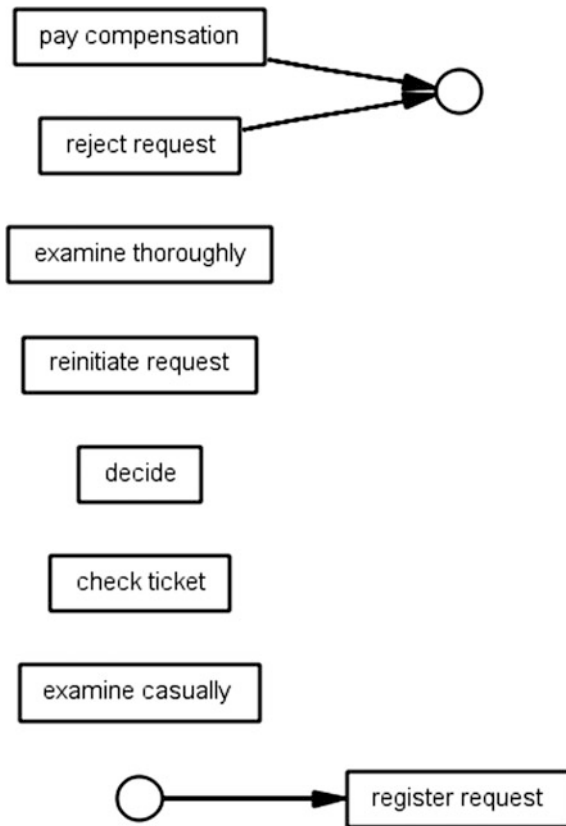


Fig. 5. Tsinghua-Alpha algorithm

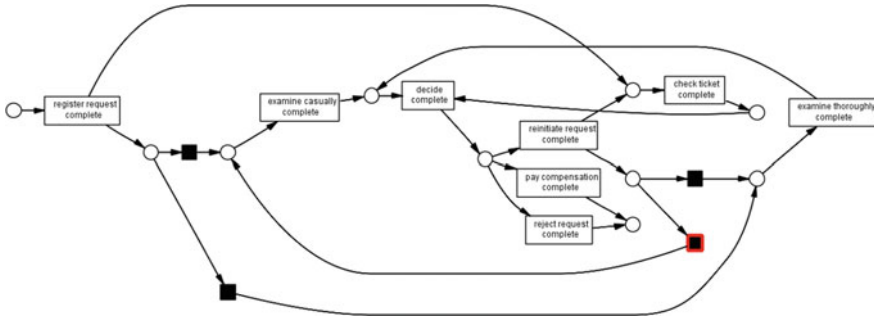


Fig. 6. GAE

5 Results and Outcomes

In this, we received classified result for that particular hospital case. We got 8 classified event classes. For this classification, we used 2 criteria, i.e., on the basis of event functionality and other one on the basis of event type. Along with this, complete classification is further subdivided into 3 parts, i.e., start event (count is 1), originators (count is 6) and end event (count is 2). After interpreting the Petri net, we find that which workflow model we need to work upon to improve the process of information system. Please find below the outcomes (Table 5).

Table 5. Classified event log data

Model element	Event type	Occurrences (absolute)	Occurrences (relative) (%)
Check ticket	Complete	9	21.429
Decide	Complete	9	21.429
Register request	Complete	6	14.286
Examine casually	Complete	6	14.286
Reinitiate request	Complete	3	7.143
Examine thoroughly	Complete	3	7.143
Pay compensation	Complete	3	7.143
Reject request	Complete	3	7.143

6 Conclusion and Future Work

“Workflow scientist” desires to possess particular/exact to initiate innovation in a progressively digitalized ecosphere. In this paper, we just only conceptualized our idea with a small case study. This allows us to analyze the operational process workflow of healthcare information systems under real-life scenarios, and use extraction procedures for processes to acquire specific and recognized software improvement models. This paper oriented on associating the organized mining approach for the event classes from event data for the software process improvement by using the Petri nets flow model

approach. In the future, we aim at conducting additional experiments using different variety of event log data sets. A reasonable succeeding phase is to progress with tool support for domain-based information management systems.

References

1. van der Aalst WMP (2011) *Process mining: discovery, conformance and enhancement of business processes*. Springer, Berlin
2. Adriansyah A, van Dongen BF, van der Aalst WMP (2011) Towards robust conformance checking. In: *Business process management workshops. Lecture notes in business information processing*, vol. 66, pp. 122–133. Springer, Berlin, Heidelberg
3. Dehnert J, van der Aalst WMP (2004) Bridging the gap between business models and workflow specifications. *Int J Coop Inf Syst* 13(3):289–332
4. Schimm G (2004) Mining exact models of concurrent workflows. *Comput Ind* 53(3):265–281
5. Goedertier S (2008) *Declarative techniques for modeling and mining business processes*. Phd thesis. Katholieke Universiteit Leuven, Faculty of Business and Economics, Leuven, Sept
6. Pesic M, van der Aalst WMP (2006) A declarative approach for flexible business processes management. In: Eder J, Dustdar S (eds), *Business process management workshops. Lecture notes in computer science*, vol 4103, pp 169–180. Springer
7. Günther CW XES standard definition. <http://www.xes-standard.org>
8. Agrawal R, Gunopulos D, Leymann F (1998) Mining process models from work logs. In: Ramos I, Alonso G, Schek H-J, Saltor F (eds), *Advances in database technology—EDBT'98: sixth international conference on extending database technology. Lecture notes in computer science*, vol 1377, pp 469–483
9. Pinter SS, Golani M (2004) Discovering workflow models from activities lifespans. *Comput Ind* 53(3):283–296
10. IBM (1999) *IBM MQSeries workow—getting started with buildtime*. IBM Deutschland Entwicklung GmbH, Boeblingen, Germany
11. Cook JE, Du Z, Liu C, Wolf AL (2004) Discovering models of behavior for concurrent workflows. *Comput Ind* 53(3):297–319
12. Greco G, Guzzo A, Pontieri L, Sacca D (2004) Mining expressive process models by clustering workflow traces. In: Dai H, Srikant R, Zhang C (eds), *BIBLIOGRAPHY 365 PAKDD. Lecture notes in computer science*, vol. 3056, pp 52–62. Springer
13. Herbst J, Karagiannis D (2004) Workow mining with InWoLvE. *Comput Ind* 53(3):245–264
14. Regev G, Soffer P, Schmidt R (2006) Taxonomy of flexibility in business processes, Input BPMDS'06 workshop