**1. INTRODUCTION**

Most organizations use information systems to support the execution of their business processes. Examples of information systems supporting operational processes are Workflow Management Systems (WMS), Customer Relationship Management (CRM) systems and Enterprise Resource Planning (ERP) systems and so on. These information systems may contain an explicit model of the processes that may support the tasks involved in the process without necessarily defining an explicit process model. Process mining is a relatively young research discipline that sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand. The idea of process mining is to discover, monitor and improve real processes by extracting knowledge from event logs readily available in today's (information) systems [2]. The main benefit of process mining techniques is that information is objectively compiled. In other words, process mining techniques are helpful because they gather information about what is actually happening according to an event log of an organization, and not what people think that is happening in this organization.



Fig 1: The process-mining pipeline, from executing processes

to mining and analytics[8].

Process mining evolved in the context of analyzing software engineering processesby Cook and Wolf in the late 1990s [7]. Agrawal and Gunopulos [5] introduced process mining to the context of workflow management. Major contributions to the field have been added during the last decade by van der Aalst and other research colleagues by developing mature mining algorithms and addressing a variety of topic related challenges [1].This has led to a well developed set of methods and tools that are available for scientists and practitioners.

Process mining has several goals:

* Discovering process models from event logs,
* Analyzing the conformance between a prescribed and the discovered process models,

* Improving the prescribed process models.

**2.PROCESS MINING BASICS**

**2.1.Mining Algorithms**

The main component in process mining is the mining algorithm. It determines how the process models are created. A broad variety of mining algorithms does exist. The following three categories will be discussed in more detail.

* Deterministic mining algorithms
* Heuristic mining algorithms
* Genetic mining algorithms

Determinism means that an algorithm only produces defined and reproducible results. It always delivers the same result for the same input. A representative of this category is the α-Algorithm[3]**.** It was one of the first algorithms that are able to deal with concurrency. It takes an event log as input and calculates the ordering relation of the events contained in the log.

Heuristic mining also uses deterministic algorithms but they incorporate frequencies of events and traces for reconstructing a process model. A common problem in process mining is the fact that real processes are highly complex and their discovery leads to complex models. This complexity can be reduced by disregarding infrequent paths in the models.

Genetic mining algorithms use an evolutionary approach that mimics the process of natural evolution. They are not deterministic. Genetic mining algorithms follow four steps: initialization, selection, reproduction and termination. The idea behind these algorithms is to generate a random population of process models and to find a satisfactory solution by iteratively selecting individuals and reproducing them by crossover and mutation over different generations. The initial population of process models is generated randomly and might have little in common with the event log. But due to the high number of models in the population, selection and reproduction better fitting models are created in each generation.

The outcomes of mining differ depending on the used algorithm. We use the event log displayed in Table 2 to illustrate the models created by different mining algorithms

and for getting an impression how the mining works.

**2.2. Characteristics of process model**

A process model is characterized by four parameters of quality. They are:

* **Fitness** addresses the ability of a model to replay the behavior recorded in the event log.
* **Simplicity** means that the simplest model that can explain the observed behavior should be preferred.
* **Precision** requires that the model does not allow additional behavior very different from the behavior recorded in the event log.
* **Generalization** means that a process model is not exclusively restricted to display the eventually limited record of observed behavior in the event log but that it provides an abstraction and generalizes from individual process instances.

To describe a process model different graphical representations like:

Petri net,Business process model notations and Work flow nets are available.

The graphical tool used in this report is **Petri net.**

**2.3.Petri Net**

A **Petri Net** is a collection of directed arcs connecting places and transitions. Places may hold tokens. The state or marking of a **net** is its assignment of tokens( i.e. events that may occur) to places(i.e. conditions).

For the generation of Petri Net, the tool used is ProM6.5.1.

**3. LITERATURE SURVEY**

The idea of applying process mining in the context of workflow management was first introduced by Agrawal et al. [1]. This work is based on workflow graphs. In this paper, two problems are defined: The first problem is to find a workflow graph generating events appearing in a given workflow log and the second problem is to find the definitions of edge conditions.

Cook and Wolf [2] have investigated process mining issues in the context of software engineering processes. They attempted to discover software process models from the data contained in event logs. Three methods for process discovery is described here: neural networks, a purely algorithmic approach and Markovian approach. However, the results presented are limited to sequential behavior.

Cook and Wolf in [3] extended their work to concurrent processes. Specific metrics (entropy, event type counts, periodicity, and causality) are used in this paper to discover models out of event streams.

Herbst et al. [4] addressed the issue of process mining in the context of workflow management using an inductive approach. This work is limited to sequential models.

Herbst [5] extended his work on process mining in the context of workflow management that allows for concurrency.

Weijters et al. [6] proposed a heuristic approach for process discovery. In this approach, the concept of “dependency/frequency tables” and “dependency/frequency graphs” is used. The preliminary results presented in this paper focused on issues such as noise.

Schimm [6] developed a mining tool suitable for discovering hierarchically structured workflow processes. This requires all splits and joins to be balanced.

van der Aalst et al. [8] proposed the α algorithm for process discovery using the workflow logs. For the *α* algorithm it is proven that for certain subclasses it is possible to find the right workflow model. Unfortunately, this approach has problems dealing with short loops.

van der Aalst et al. [8] proposed an extended version of the α algorithm, which is known as the α+ algorithm. The α+ algorithm is able to handle short loops.

The process mining literature review shows that process mining is applied to transactional event logs to discover business process model. The logs available in other systems have not been used for process mining. Hence we will attempt to develop process mining techniques which shall be applied to non-transactional logs.

The process mining algorithms such as α algorithm are sequential algorithms. Considering the events logs can be very large, these algorithms may be inefficient. Hence we are proposing to develop process mining algorithms which can take advantage of a parallel programming system.

The heuristic algorithms are often practical solutions to complex problems. We propose to explore heuristic algorithms which may provide practical solution to the process discovery.

**4.MOTIVATION**

Discovering process models from event logs is an important part of Process Mining. Performance improvement of computationally intensive Process Mining algorithms is an important issue due to the need to efficiently process the exponentially increasing amount of event log data. It becomes computationally intensive and time consuming for process discovery algorithms to work on ever increasing large sized event logs. Process discovery algorithms give a clear cut insight of business, helping in enhancing the current workflow practices of the organizations. To improve the current standards of workflow, organizations make use of process discovery algorithm. Thus there is a need to make the process discovery algorithms efficient enough to handle the rapidly growing size of event logs.

Alpha Miner algorithm is one of the fundamental algorithms in Process Mining consisting of discovering a process model from event logs. Our analysis of the Alpha Miner algorithm reveals that the algorithm contains independent tasks which can be split among different processors

**5.OBJECTIVE**

To study Process Mining for sequential event logs and implement Alpha algothim( till generation of foot print matrix) and generate Process flow diagram(petrinet) for the designed event log using ProM-6.5.1



Figure 2: A petri net consisting of places and transitions.

**6. ALPHA MINER ALGORITHM**

Alpha Miner is one of the most popular process mining algorithms. It was proposed by Wil van der Aalst, Ton Weijters, and Laura Maruster in 2004. It is a fundamental process discovery algorithm that extracts a process model from an event log consisting of traces and represents it as a petri net [4]. It is based on analyzing immediate successor relation between activities present in the event log. Table 1.1 shows an event log in which the activities belonging to the same Case Id appear sequentially in increasing order of timestamp.

|  |  |
| --- | --- |
| **Case Id** | **Activity** |
| 1 | A |
| 1 | B |
| 1 | D |
| 1 | E |
| 2 | A |
| 2 | C |
| 2 | E |
| 3 | A |
| 3 | D |
| 3 | B |
| 3 | E |

Table 1: Event Log

The algorithm scans through the event log and establishes the following relations between all the activities:

1. Direct Succession (a >Lb) if activity a directly precedes activity b in some trace i.e. activity a having earlier timestamp value is immediately followed by activity b which is performed just after a.

2. Causal (a→Lb) if a>b and b≯a.

3. Parallel (a║Lb) if a>b and b>a.

4. Unrelated (a#Lb) if a≯b and b≯La.

Let the event log L shown in Table 1.1 is over T activities or transitions. The

detailed steps involved in the Alpha Miner algorithm are as follows:

1. TL = { t ∈ T | ∃σ ∈L t ∈σ }

Determine the set of unique activities present in the event log. In event log L, TL from T is {A, B, D, E, C}.

1. TI = { t ∈ T | ∃σ ∈L t = first(σ) }

From the set of all activities T, determine the set of activities which do not have immediate predecessor anywhere in any of the trace in the log. In event log L, TI is {A}.

1. To = { t ∈ T | ∃σ ∈L t = last(σ) }

From the set of all activities T, determine the set of activities which do not have immediate successor anywhere in any of the trace in the log. In event log L, TO is {E}.

1. Scan through the traces present in the sequential event log and determine the above mentioned relations (>,→,║,#) between all activities and represent them in the form of a matrix called footprint. Table 1.2 shows footprint matrix for L.
2. XL = {(A,B) | A ⊆ TL ∧ A ≠ ∅∧ B ⊆ TL ∧ B ≠ ∅∧∀a ∈ A ∀b∈ B a→Lb ∧∀a1,a2 ∈ A a1#La2 ∧∀b1,b2 ∈ B b1#Lb2 }

Using the footprint matrix we generate XL that consists all possible pairs of sets (A,B) such that activities within set A and within set B are unrelated to rest of the activities of their set and each activity in set A is in causal relation with every activity of set B. Table 1.3 reports XL for L.

1. YL = {(A,B) ∈ XL | ∀(A’,B’) ∈ XL A ⊆ A' ∧ B ⊆ B' ⇒ (A,B) = (A',B') } In XL, if for a set pair (A,B), all activities in A are a subset of activities in set A' and all activities in set B are a subset of activities in set B' and (A', B') set pair is present in XL, then set pair (A,B) in XL is considered to be same as set (A',B'). Table reports 1.4 YL for L.
2. PL = {P(A,B) | (A,B) ∈ YL } ∪ (iL, oL)

In PL, a place is generated for each distinct pair of set (A, B). Along with it, an input place iL and output place oL are generated.

1. FL = { (a, P(A,B)) | (A,B) ∈ YL ∧ a ∈ A } ∪ { (P(A,B), b) | (A,B) ∈ YL ∧b ∈ B} ∪ { (iL, t) | t ∈ TI} ∪ {(t, oL) | t ∈ TO }

For each set pair (A, B) in YL, arcs are connected from every activity present in set A to a place generated for the set pair (A, B) and arcs are also connected from the place to every activity present in set B. Activities in TI are connected to the input place iL and activities in TO are connected to the output place oL.

1. α(L) = (PL, TL, FL) The generated petri net of Alpha Miner algorithm is represented by PL, TL and FL as shown in Figure 1.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **D** | **E** | **C** |
| **A** | # | → | → | # | → |
| **B** | ← | # | ║ | → | # |
| **D** | ← | ║ | # | → | # |
| **E** | # | ← | ← | # | ← |
| **C** | ← | # | # | → | # |

Table 2: Footprint Matrix

|  |  |
| --- | --- |
| **Set A** | **Set B** |
| A | B |
| A | D |
| A | C |
| B | E |
| D | E |
| A | {B,C} |
| A | {D,C} |
| C | E |
| {B,C} | E |
| {D,C} | E |

|  |  |
| --- | --- |
| **Set A** | **Set B** |
| A | {B,C} |
| A | {D,C} |
| {B,C} | E |
| {D,C} | E |

Table 3: Set pair (A, B)

Table 4: Maximal Set pair (A, B)

**7. RESEARCH FRAMEWORK AND SOLUTION APPROACH**

To generate process flow diagram for sequential event log,a synthesized event log was taken.Alpha algorithm was implemented.The petrinet was generated using ProM- 6.5.1.

The first step of alpha algorithm is finding the set of unique activities present in the sequential event log. For this step, each activity is to be compared with all the other activities present in the entire event log.The second step is to find the set of all starting events in the entire event log.The third step is to find the set all end events and the fourth is the generation of foot print matrix.

**8.EXPERIMENTAL SETTINGS AND RESULTS**

A sequential,synthesised event log was taken with three fields (case id,events,timestamp) with a size of 2000\*3. The alpha algorithm was implemented on it using C programming and the foot print matrix was generated.In order to generate the petri net ProM-6.5.1 was used. The generated petri nFig 2. Snapshot of the petrinet generated in ProM-6.5.1.

**9.CONCLUSION**

Process Mining for sequential event logs was studied and Alpha algorithm was used for process discovery. The input for the alpha algorithm was an event log and the output was a petri net showing the discovered process model.The petri net was generated using ProM6.5.1.

**10.FUTURE WORK**

Discovering process models from event logs is an important part of Process Mining. Performance improvement of computationally intensive Process Mining algorithms is an important issue due to the need to efficiently process the exponentially increasing amount of event log data. It becomes computationally intensive and time consuming for process discovery algorithms to work on ever increasing large sized event logs. Process discovery algorithms give a clear cut insight of business, helping in enhancing the current workflow practices of the organizations. To improve the current standards of workflow, organizations make use of process discovery algorithm. Thus there is a need to make the process discovery algorithms efficient enough to handle the rapidly growing size of event logs.

Alpha Miner algorithm is one of the fundamental algorithms in Process Mining consisting of discovering a process model from event logs. Our analysis of the Alpha Miner algorithm reveals that the algorithm contains independent tasks which can be split among different processors or threads and the algorithm has the ability or property of parallelization.

**REFERENCE**

1. Arawak, R., Gunopulos, D., and Leymann, F. 1998, “Mining Process Models from Workflow Logs,” In Proc. Sixth Int’l Conf. Extending Database Technology, pp. 469–483.

2. Cook, J. E., and Wolf, A. L. 1998a, “Discovering models of software processes from event-based data,” ACM Trans. Softw. Eng. Methodol. (7:3), pp. 215–249.

3. Cook, J.E. and Wolf, A.L. 1998b, “Event-Based Detection of Concurrency”, In Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6), pages 35–45.

4. Herbst, J. 2000, “Dealing with Concurrency in Workflow Induction”, In U. Baake, R. Zobel, and M. Al-Akaidi, editors, European Concurrent Engineering Conference. SCS Europe.

5. Herbst, J. and Karagiannis, D. 2000, “Integrating Machine Learning and Workflow

Management to Support Acquisition and Adaptation of Workflow Models”, International Journal of Intelligent Systems in Accounting, Finance and Management, 9:67–92.

6. Schimm, G. 2002, “Process Miner - A Tool for Mining Process Schemes from Event based Data”, In S. Flesca and G. Ianni, editors, Proceedings of the 8th European Conference on Artificial Intelligence (JELIA), volume 2424 of Lecture Notes in Computer Science, pages 525–528. Springer-Verlag, Berlin.

7. Tiwari, A., Turner, C. J., and Majeed, B. 2008, “A review of business process mining: state-of-the-art and future trends,” Business Process Management Journal (14:1), pp. 5–22.

8. van der Aalst, W. M. P. 2011, “Process Mining: Discovery, Conformance and Enhancement of Business Processes”, (1st Edition) Berlin; Heidelberg: Springer.

9. van der Aalst, W. M. P., Andriansyah, A., Alves de Medeiros, A. K., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., Van den Brand, P., Brandtjen, R., and Buijs, J. 2012, “Process mining manifesto,” In BPM 2011 Workshops Proceedings, pp. 169–194.

10. van der Aalst, W. M. P., Weijters, A. J. M. M., and Maruster, L. 2002a, “Workflow Mining: Which Processes can be Rediscovered,” (Vol. 2480) Presented at the Proc. Int’l Conf. Eng. and Deployment of Cooperative Information Systems (EDCIS 2002), pp. 45–63.

11. van der Aalst, W.M.P. and van Dongen, B.F. 2002b, “Discovering Workflow Performance Models from Timed Logs”, In Y. Han, S. Tai, and D.Wikarski, editors, International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002), volume 2480 of Lecture Notes in Computer Science, pages 45–63. Springer-Verlag, Berlin.

12. Weijters, A.J.M.M. and van der Aalst, W.M.P. 2001, “Process Mining: Discovering Workflow Models from Event-Based Data”, In B. Kr¨ose, M. de Rijke, G. Schreiber,and M. van Someren, editors, Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), pages 283–290.

13. Weske, M. 2012, “Business process management concepts, languages, architectures”, New York: Springer.