Enhancing Business Process models by Optimizing resource allocation

Tati Pati Sai Krishna
Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
sai.tatipati@gmail.com

Dr. Emmanuel.M
Head, Dept. of Information Technology
Pune Institute of Computer Technology
Pune, India
emman2001@gmail.com

***Abstract*—Organizations today strive for a high return on their investments. Every organization has a business process model on which it works. These models depict the way the business runs. A large amount of data is stored in the event logs which are audits of the model that an entity of the organization works on. The process mining techniques available today allow us to discover, analyze and optimize the business models based on the knowledge from the event logs. Recently, the IEEE task force on process mining released the process mining manifesto which includes the various challenges that process mining faces. For a high return of investment, the primary objective for any organization is to optimize its business process model. The optimization is heavily dependent on the optimization of resources that an organization possesses. In this paper, we will exclusively study the resource allocation in a business process model and device a model for optimum utilization of these resources.**

Keywords-Business Process Managemnt, Process mining, process discovery, Business process model notation, Business process model optimization.

I. INTRODUCTION

Business Process Management is an important field in operations management that focuses on improving corporate performance by managing and optimizing a company's business processes. Today the organizations prefer a high return of interest for the efforts and capital they put in. The organization's business processes are defined using BPM tools like IBM BPM, TIBCO AMX-BPM etc. The performance of the business processes depends on the model that is developed using the various BPM tools available. These tools maintain a database of the activities that are executed as per the business processes. The information is stored the database in the form of "event logs". The event logs are a great place for knowledge discovery related with the business processes. Organizations and even governments of different countries are investing a lot on the research of improving the process models using the knowledge from the event logs. Process mining is the nascent research field at the intersection of Business process management and data mining. The knowledge discovered from the event logs can be used to provide suggestions and comments to the analysts who can then improve the business models. The event logs can also be used to discover knowledge related with the resources and their allocation as well as utilization. This will help in improve the efficiency of resources by increasing or decreasing their utilization. Process mining has been around for a while now but the importance and induction of process mining to the businesses is of recent origin. Process mining covers three main perspectives, the process perspective, the organization perspective and the case perspective. All the three perspectives have to be studied and knowledge from all the three perspectives is important to improve and optimize the business processes.

To achieve optimization, we will study and compare various process discovery algorithms and then implement one or more algorithms that work best for real-life event logs. We will also study and implement one or more resource allocation and resource utilization algorithms for resource optimization.

II. RELATED WORK

. Process mining has been applied in many areas including software engineering. Software engineering practitioners are constantly developing improved ways to develop quality software in an efficient way. However, assuring that the process defined is actually followed by the development team is usually a difficult task. In order to better control their projects, software houses usually automate their development processes through a monitoring system. This system is capable of registering valuable information about process execution, such as timestamps, task identification, stakeholders involved in the execution of an activity etc., in the form of execution logs. Such execution logs (also called event logs or audit trails) are the starting point for the process mining technique, which was developed to semi automatically extract useful information from them.

Wil van der Aalst, discussed process mining in the context of services[1]. The author used the guiding principles and challenges described in process mining manifesto to describe the state-of-the-art in process mining. The author called the techniques of process mining for web-based, loosely coupled services as service mining. He concluded that Service mining is concerned with 1) the discovery of service behaviour, 2) checking conformance of services, and 3) service model extension (e.g., showing bottlenecks based on event data). Finally, the author concludes that despite the huge potential of service mining, additional research is needed to improve the applicability of process mining in distributed and loosely coupled systems.

Weijters A.J.M.M et al, introduced a new process modelling language (e.g. Causal Matrices) and illustrated the advantages of Causal Matrices during the mining of hidden activities[2]. The authors provided a new algorithm called as HeuristicsMiner which can be applied for real-life event logs. They also proved that in practical situation (i.e. with event log with thousands of traces, low frequent

behavior and some noise) the HeuristicsMiner can focus on all behavior in the event log, or only the main behavior. The only drawback of HeuristicsMiner algorithm is that of with non-local dependencies. HeuristicsMiner algorithm may under-perform when the event logs contain non-local dependencies.

Jochen De Weerdt et al, carried out a multi-dimensional evaluation study was in order to evaluate process mining techniques comprehensively on eight real-life event logs[3]. The authors claim that there exists an important difference between evaluation of process discovery algorithms based on either artificial or real-life event logs. The authors concluded that HeuristicsMiner seemed to be the most appropriate and robust technique in a real-life context in terms of accuracy, comprehensibility, and scalability. They also statistically validated that dealing with high complexity event logs remains an important challenge for process mining algorithms, both in terms of accuracy as well as in terms of comprehensibility. Finally they suggested that the future of process mining research should emphasize on developing insightful techniques for analyzing real-life event logs.

The foundational approaches to process discovery were formulated by Agrawal et al, Cook and Wolf, and Datta et al. Cook and Wolf proposed three different approaches in the context of software engineering[4]. The α-algorithm can be considered as one of the most substantial techniques in the process mining field. Roziant et al proved that it can learn an important class of workflow nets, called structured workflow nets, from complete event logs[5]. The α-algorithm assumes event logs to be complete with respect to all allowable binary sequences and assumes that the event log does not contain any noise. Therefore, the α-algorithm is sensitive to noise and incompleteness of event logs.

Artini M. Lemos et al, demonstrated that process mining can be a useful tool for making the auditing activity, which is conducted for checking the conformance of the actual development process to company's developed process model, less costly and more efficient[6]. The authors also analysed the opportunities provided by process mining as a software development management tool, and found several important applications. The authors compared process mining features with the requirements of the well-known Capability Maturity Model (CMMI) and suggested the importance of process mining with CMMI model.

III. PROCESS MINING

Process mining techniques enables an analyst to discover process models and analyze them using the knowledge available in the event logs. Once analyzed the knowledge from the event logs can be used to optimize the model. Process mining techniques broadly fall under three major categories[7]; Process discovery, conformance checking and model enhancement, and we describe them in this section.

A. Process discovery

As discussed earlier, an event-log is the starting point for process mining. All the knowledge is stored in a discrete from in this so called event –log. An event log can be seen as consisting of various events. A simple example of an event may be shown as follows:

Activity id : 1
Resource : Jake
Concept-name : CheckForFunds
Time-stamp : 13-1-15 22:00 IST
Status : Complete

Depending on the business application, there may be thousands if not millions of such events in the event log. The events essentially capture the behavior of the business process. The event logs can be stored in a database on the client machine or can be stored remotely in a server. The event log can be stored as a relational table or as a non-relational table(Using No-SQL database).

All the parameters related with an event can be used for process mining. But we use the activity id and the concept-name for discovering the business process model. Together they are referred to as a case. In [1] an event log is described as a multiset of traces, where each trace belongs to a sequence of activities. Assuming a model consisting of a set of activities, {sta,cff,cfr,pto}, an example trace can be shown as <sta,cff,cfr>. Similarly, an event log L can de shown as L = [$\langle sta,cff,cfr\rangle^{20}$, $\langle sta,pto,cfr\rangle^{15}$,$\langle sta,cfr\rangle^{10}$,$\langle sta\rangle^{5}$]. In this event log, there are three kinds of traces with a total of 50 traces. For example, there are 20 traces that follow <sta,cff,cfr>. The goal of process discovery algorithms is to discover a process model from this kind of event logs. For example the alpha algorithm generates a petri net, a HeuristicsMiner algorithm generates a causal matrix. As we can observer, the events refer to the concept-names. Hence, the process discovery algorithms can analyze the concept-name to find the activity. One could argue that, what is the need of discovering the process models from the event logs when the models can be generated from the source code or documentation. The autonomous nature of services may tend to change the nature of the process model over a period of time. Hence, it is necessary to identify the model from the event log. Also, the discovered model form the basis for a lot of analysis in the next steps of the process mining.

B. Conformance Checking

The fact that makes conformance checking interesting is that, the behavior of a process model may change during the run time. That is, the actual behavior as portrayed by the event logs, may differ from the one that is modelled. Conformance checking aims to discover the discrepancies in the modelled behavior and the actual behavior. For example in our event log, the application is started 5 times, but it is never moved forward. That is, the application had to be forcefully deleted. This is generally a bad sign for the process model, as the business should essentially close all process instances. If there are a large number of traces in the log that are impossible to get executed according to the modelled behavior, then, we say that the conformance is poor. For example, if we obtain traces like <cfr,pto> or <cff,pto> without even starting the application, we can confirm that these traces are noise in

the event log and doesn't really depict the model's behavior.

Conformance can be viewed from two angles: 1) the model does not capture the real behavior and 2) reality deviates from the desired model. Conformance checking is very important for compliance checking, auditing, certification, and runtime monitoring[1]. Typically four quality dimensions are considered for comparing model and the log, they are 1) fitness, 2) simplicity, 3) precision and 4) generalization[8]. Obviously the simplest model that can explain the behavior seen in the log is the best model. This principle is called as Occam's Razor.

C. *Model Enhancement*

The process of improving an existing model using the knowledge from the event log is called model enhancement. A non-conforming model can be corrected using the analysis gathered from the event log. Resource optimization plays an important role for enhancing the overall model. The information related with the resources in the organization and their assignment to various activities provides a good arena for optimizing the business process models of the organization[9]. Simply by measuring the time difference between causally related events and computing the basic statistics such as averages, variances and confidence intervals, it is possible to identify main bottlenecks.

With the analysis of the event-logs it is possible to gain insights into the way the activities are assigned to various resources. The event-logs can also be used to measure the resource performance. Process mining can also be used to provide suggestions, recommendations and predictions at run time. Model enhancement thus, deals with improving the process model and optimizing the resources.

IV. PROPOSED SYSTEM

We propose a system which consists of a model generator and an optimizer. The model generator basically does the job of process discovery and conformance checking. The event log is the starting point for the process mining and it can be input to the system in two ways. 1) The analyst can import the event log and 2) Automatically the tool can import the event log from the related databases. Once the event log is imported, a process model is discovered, we call it the intermediate process model. This model is used for conformance checking. In this stage, the pitfalls like deadlocks, allocation of resources etc., are identified. After this a final model based on the modelled behavior and observed behavior is generated as the output of the model generator.

The second part of our model is the optimizer. The knowledge discovered in the generator module along with the discovered model is passed on to the optimizer. The optimizer first generates a task operation model where the aim is to find the resources that are mapped with the activities. Once the task operation model is generated different permutations are tried and a path with higher efficiency is obtained.

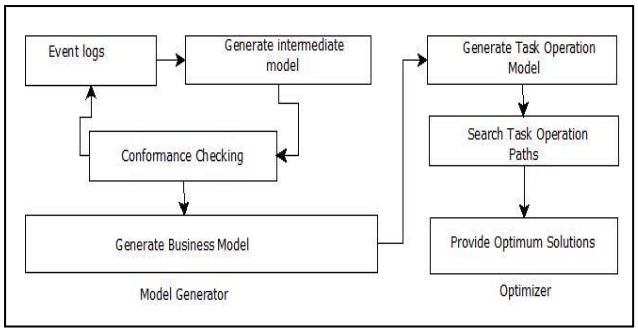

Figure 1. The proposed model for process mining

V. IMPLEMENTATION DETAILS

The HeuristicsMiner algorithm as given in [2] mines the control flow perspective of a process model. To do so, it only considers the order of the events within a case. In other words, the order of events among cases isn't important. The timestamp of an activity is used to calculate these ordering. It is important that for any case, say, case 1 activity A is followed by B within the context of case 1 and not that activity A of case 1 is followed by activity A of another case, say, case 2. Therefore, the event log is defined as follows. Let A be a set of activities. $\sigma \in A^*$ is an event trace., i.e., an arbitrary sequence of activity identifiers. $W \subseteq A^*$ is an event log, i.e., a multiset of events. An example of an event log can be given as W = [KLMN, KMN, KLN, KN] where K,L,M,N are individual activities. To find a process model on the basis of an event log, the log should be analyzed for causal dependencies, e.g., if an activity is always followed by another activity it is likely that there is a dependency relation between both activities.

Let W be an event log over A, i.e., $W \subseteq A^*$. Let $k, l \in A$:

1) $K >_w l$ iff there is a trace $\sigma = t_1 t_2 t_3 \ldots t_n$ and $i \in \{1,\ldots,n-1\}$ such that $\sigma \in W$ and $t_i = k$ *and* $t_{i+1} = l$,
2) $k \rightarrow_w l$ iff $k >_w l$ *and* $l \not>_w k$.
3) $k \#_w l$ iff $k \not>_w l$ *and* $l \not>_w k$.
4) $k \|_w l$ iff $k >_w l$ *and* $l >_w k$.
5) $k >>_w l$ iff there is a trace $\sigma = t_1 t_2 t_3 \ldots t_n$ and $i \in \{1,\ldots,n-2\}$ such that $\sigma \in W$ and $t_i = k$ *and* $t_{i+2} = l$

The first relation $>_w$ describes which activities appeared in sequence. Relation $\rightarrow_w$ is called the direct dependency relation derived from the event log W. Relation $\|_w$ suggests concurrent behavior, i.e., potential parallelism. If two activities can follow each other directly in any order, then they are likely to be in a parallel relation. Relation $\#_w$ gives pairs of transitions that never follow each other directly. The main idea in HeuristicsMiner is to take the frequency of the above mentioned relations into account while deciding on the process model. This technique is less prone to noise and incompleteness of logs.

A. *Mining for the model*

A frequency based metric is used to indicate how certain we are that that there is truly a dependency relation between two events K and L. This metric is used in heuristic search for the correct dependency relations.

Let W be an event log over T, and $k,l \in T$, and $k,l \in T$. Then, $| k >_w l |$ is the number of times $k >_w l$ occurs in W, and

$$k \Rightarrow w\ l = \left(\frac{| k >w l | - | l >w k |}{| k >w l | + | l >w k | + 1}\right) \quad (1)$$

It can be seen that, the value of $k \Rightarrow w\ l$ is always between -1 and 1. The effect of this method can be seen if the number of traces are large. For example if there are 5 traces in which l always follows k then the value of $k \Rightarrow w\ l$ becomes 5/6 = 0.83. But for 100 traces in which l always follows k and the vice versa is never followed then, the value of $k \Rightarrow w\ l$ becomes 100/101 = 0.99. A high $k \Rightarrow w\ l$ value indicates that we are pretty sure of a dependency relation between k and l. In a practical situation we never know if any trace is really a noise or if it is a low frequent pattern. To handle this, three parameters are available in the HeuristicsMiner: i) the Dependency threshold, ii) the Positive observations threshold, iii) the Relative to best threshold. With these threshold values, we can accept a dependency if, a) there is a dependency measure above the value of the Dependency threshold, and ii) have a frequency higher than the value of the Positive observations threshold, and iii) have a dependency measure for which the difference with the "best" dependency measure is lower than the value of Relative to best threshold.

Short loops can be mined using the HeuristicsMiner using the following equation.

$$k \Rightarrow w\ k = \left(\frac{| k >w k |}{| k >w k | + 1}\right) \quad (2)$$

The AND/XOR-split can also be derived from the HeuristicsMiner. This is because of the use of Causal matrix. The following measurement is defined to express the above relations.

$$k \Rightarrow w\ l \wedge m = \left(\frac{| l >w m | + | m >w l |}{| k >w l | + | k >w m | + 1}\right) \quad (3)$$

VI. EXPERIMENTAL SETUP AND RESULTS

The starting point for process mining is always the event log. The event log that we are using for our experimentation is downloaded from the process mining website. The event log is in the form of an .xes file. It is similar to a XML file and can be viewed as an XML file. Figure 1 shows a part of the event log.

Figure 2. A single trace of the event log.

The event log consists of 503 traces with 1848 repeated activities. Each trace consists of a sequence of activities which are performed. Each activity can be performed by the same resource or different resources. In our event log, we see that there are four resources, John, Terry, Lily and Rosy. Although for the process discovery, the knowledge of resources doesn't matter much but for calculating resource utilization and resource conflicts, the knowledge of available resources is of utmost importance. The logic behind understanding the different activities in the log is by understanding the XML file. From the above figure we can see that, every activity is defined under the <event> tag. The <string> tag inside the event tag is of importance as the attributes of <string> tag provide us with all the important information such as the activity name by the key "concept:name", resource by key "org:resource: and the like. We try to parse this XML file using any XML parser (Here we have used document parser). Using the XML parser, we parse the XML file for <string> tag inside the <event> tag for all the events for all the traces under <trace> tag. Thus, we come to know that there are 1848 activities in the event log with 503 traces.

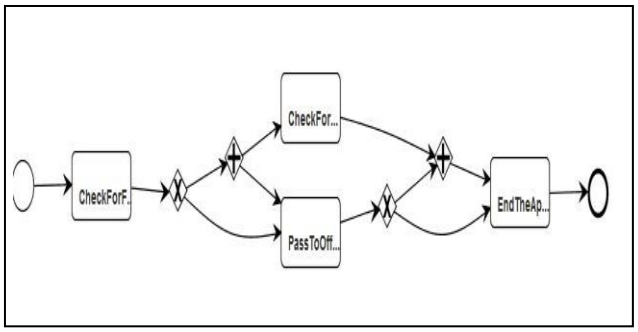

Figure 3. Model Generated using HeuristicsMiner in ProM

Figure 2 shows the model that is generated using the HeuristicsMiner algorithm which is available as a plugin in ProM framework. The same event log is mined for discovering the model.A single trace of the event log.

Figure 3 shows the output of our java program that implements the HeuristicsMiner algorithm and also

analyses the resource utility and conflicts if any as represented by the event lot.

The output shows that the model generated is very much similar to what the simulated model is and also the knowledge about the resources as available in the event log. The results in figure 3 also show the loads on the various resources and comments on the loads if any required.

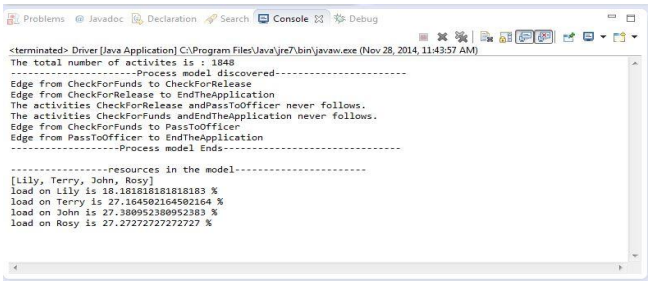

Figure 4. Model obtained and resource allocation.

Figure 4 shows the graph that is generated for the resource allocation and utilization. The graph shows the mined information about the loads on various resources. From the graph it can be derived that what resources are equally-balanced or under loaded.

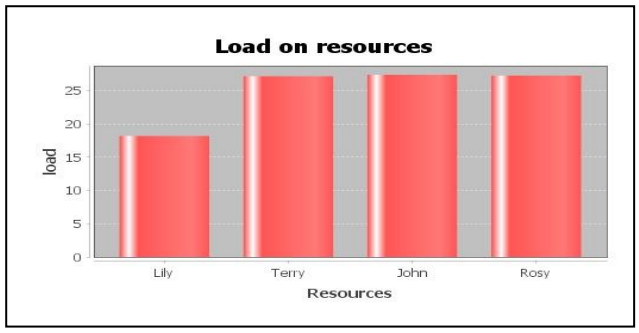

Figure 5. Graph showing utilization of available resources

Further, we would be taking different event logs and applying process mining techniques on those to obtain other results. The results will include generation of reports and providing insights to the project managers regarding deadlocks in the process models, over loading of particular resource etc.

ACKNOWLEDGMENT

We would like to thank Mr. Shirish Phanse and Mr. Gaddam Srinivas from TIBCO software Inc. for their constant support during this work.

REFERENCES

[1] Wil Van Der Aalst, "Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior", IEEE Transactions on service computing, Dec, 2013.

[2] A.J.M.M Weijters, W.M.P. Van der Aalst and A.K. Alves de Medeiros, "Process mining with the Heuristics Miner algorithm", Eindhoven University of Technology press, Dec, 2006

[3] Jochen De Weerdt, Manu De Backer, Jan Vanthienen and Bart Baesens "A Multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs", Elsevier, Nov, 2011

[4] Aubrey J Rembert and Clarence Ellis, "An initial approach to mining multiple perspectives of a business process", TAPIA'09: The fifth Richard Tapia celebration of diversity in computing conference, ACM, New York, NY, USA, 2009,.

[5] A Rozinat, R.S. Mans, M.Song and W.M.P van der Aalst, "Discovering simulation models", Information systems, Nov, 2009.

[6] Artini M. Lemos et al, "Using process mining in software development process management: A case study", IEEE, 2011.

[7] Wil Van Der Aalst, "Process Mining:Discovery, Conformance and Enhancement of Business processes", Springer, Verlag, Berlin(ISBN 978-3-642-19344-6)

[8] Wil M.P. van der Aalst , Schahram Dustdar, "Process Mining manifesto", Proc. Business process management workshop, IEEE Task Force on Process Mining. 2011.

[9] Zhengxing Huang, Xudong Lu, and Huilong Duan, "A Task Operation Model for Resource Allocation Optimization in Business Process Management", IEEE transactions on systems, man, and cybernetics, Sept, 2012.

.