Constructing Workflow Clusters Based On Appropriateness Conformance Checker

O. M. Hassan, M. S. Farag
Department of Mathematics, Facility of Science
Al-Azhar University
Cairo, Egypt
{ohassan, farag.sayed}@azhar.edu.eg

Abstract: - Process mining techniques attempt to extract non-trivial and useful information from event logs recorded by information systems. Process mining techniques have recently received notable attention in the literature for their ability to assist in the (re)design of complex processes by automatically discovering models that explain the events registered in some log traces provided as input. Real-life processes tend to be less structured and more flexible. An approach to overcome this is to cluster process instances such that each of the resulting clusters corresponds to coherent sets of process instances that can each be adequately represented by a process model. On the other hand the conformance checker methods check if model and the log conform to each other or not. This paper proposed an approach to use Appropriateness Conformance Checker methods to split the event log into homogeneous subsets and for each subset a process model is created. To illustrate this we present a real-life case study from reality mining dataset provided by MIT (Massachusetts Institute of Technology) Media Laboratory. The whole approach has been implemented in ProM the process mining framework.

Key-Words: - Process mining, workflow mining, reality mining, process clustering, conformance checker.

1 Introduction

The goal of process mining is to reverse the process and collect data at runtime to support workflow design and analysis [1]. Process mining aims at extracting information from event logs to capture the process as it is being executed, so the main benefit of process mining techniques is that information is objectively compiled [2].

Conformance check means to weigh the “distance” between the behavior described by the process model and the behavior actually observed in the workflow log. But there is another interesting dimension of conformance; a “good” process model should somehow be minimal in structure to clearly reflect the described behavior. Appropriateness conformance checker consist of structural appropriateness and behavioral appropriateness. Appropriateness totally represents degree of accuracy in which the process model describes the observed behavior, combined with the degree of clarity in which it is represented [3].

Process mining techniques can deliver valuable, factual insights into how processes are being executed in real life. Real-life processes tend to be less structured and more flexible. Traditional process mining algorithms have problems dealing with such unstructured processes and generate spaghetti-like process models that are hard to comprehend. An approach to overcome this is to cluster the traces such that each of the resulting clusters corresponds to a coherent set of cases that can be adequately represented by a process model [4].

Our approach in this paper is to use appropriateness conformance checker methods as a distance measure method to clustering process model. We apply our approach to analysis communications information in reality mining dataset provided by MIT Media Laboratory [5-6]. We implement this approach in ProM framework as mature tools that were developed to support the various forms of process mining [7-8].

This paper structure as follows: Section 2 introduces a running example that will be used to illustrate our approach. Section 3 provides an overview about process mining and Conformance checker methods. Section 4 discusses how we use appropriateness conformance checker to cluster process model. Section 5 presents related work. Finally, Section 6 concludes with remarks on future work.

2 Running Example

The example that we will be used in this paper represents the process of selected communication data from reality mining dataset. The Reality Mining project introduced by MIT Media Laboratory to study followed 94 subjects using mobile phones preinstalled with several pieces of software that recorded and sent the researcher data about call logs, Bluetooth devices in proximity of approximately five meters, cell tower IDs, application usage, and phone status[5-6]. The collected...
information by 94 human subjects over the course of the academic year represent approximately 450,000 hour of information about users’ location, communication and device usage behavior[9].

We focus on communications events data that represent behavior using mobile phones. In reality mining dataset the communications events data form “for each volunteer” as following: (TIME) 20060720T211505 (DESCRIPTION) Voice call (DIRECTION) Outgoing (DURATION) 23 (NUMBER) 6175559821.

We present one day events as a single process instance, this process instance starts with “Start Day” audit trial and end with “End Day” audit trail. The others available audit trials for each instance consisting of combination between the description and direction.

3 Process Mining And Conformance Checker

Instead of starting with a process design, process mining starts by gathering information about the processes as they take place. For any information system using transactional systems or PAIS (Process Aware Information System) such as ERP (Enterprise resource planning), CRM (Customer relationship management), B2B (Business-to-business) and WFM (workflow management) systems will offer information about the order in which the events of a case are executed [2]. This information called “Event Log” and this the start point of process mining.

Process mining uses the information available in this event log to reconstruct the order of activities in the form of a graphical model (i.e. process model). The model represents the executed processes based on the logs. There are three classes of process mining techniques based on whether there is an a priori model or not:
1) Discovery: There is no a priori model and based on an event log we constructed the model.
2) Conformance: There is an a priori model. This model is compared with the event log.
3) Extension: There is an a priori model. This model is extended with a new aspect or perspective.

In process mining there are several techniques to discover process model. The following three different examples of process mining techniques:
1) Alpha Mining [10]: this algorithm works based on local strategy technique to build model. The alpha algorithm assumes event logs to be complete and does not contain any noise. Therefore, the alpha algorithm is sensitive to noise and incompleteness of event logs.
2) Genetic Mining [11]: algorithm works based on global strategy technique to build model. This technique can deal with noisy and duplicate tasks and can provide us with detailed model.
3) Heuristics mining [12]: this technique extend alpha algorithm by consider the frequency of traces in the log. Heuristics miner can deal with noise, and can be used to express the main behavior.

Process mining techniques provide a powerful means to analyze existing business processes on the basis of the actual execution logs. Based on the event log a process model can be derived, reflecting the observed behavior and therefore providing insight in what actually happened. In contrast, it is very often the case that there is already a model available, defining how the process should be carried out. Together with the data recorded in the log, this raises the interesting question “Do the model and the log conform to each other?”[13].

Conformance testing, or conformance analysis, aims at the detection of inconsistencies between a process model and its corresponding execution log, and the quantification of the gap. There are two dimensions of conformance. The first dimension is fitness, which can be characterized by the question “Does the observed process comply with the control flow specified by the process model?”. The second is appropriateness, which can be associated with the question “Does the model describe the observed process in a suitable way?”. Furthermore, this suitability must be evaluated from both a structural and a behavioral perspective [13]. Current conformance metrics work on Petri Net [14] model language.

We focused on Improved appropriateness metric “a” [13] as high evaluated method to test conformance between model and event log. Improved appropriateness metric consists of two metric, first one is improved structural appropriateness metric “a’S” and second is improved behavioral appropriateness “a’B”. And finally improved appropriateness metric is:

\[
a = a’_S \cdot a’_B \tag{1}
\]

4 Clustering Based On Appropriateness Conformance Checker

This section shows the approach of using conformance checker to clustering process model. We built our approach based on Hierarchical clustering [15] and modify it to generate only one clustering level. Appropriateness percentage is main user parameter that determines how far clustered model should be appropriateness with corresponded event logs and based on this parameter any single audit trail in event log will accepted or rejected to join a cluster. Conformance checker drilling down on extracted model to cover all cases in event log. Although this heavy analysis is limited for any cases but it may be exhausted the
available resources, so we use the maximum depth level parameter to limit the appropriateness conformance checker drilling down to avoid resources exhausted.

We use heuristics mining [12] technique to extract model from even logs. Heuristics miner provide result in Heuristic Net [7] as modeling language. But appropriateness conformance checker working on Petri Net modeling language, so we use conversion utility exist in ProM to convert Heuristic Net to Petri Net [7].

As shown in Fig. 1. The algorithm starts by iterating on all audit trials in even log. And for each audit trail decides that this audit trail belong to pre constructed cluster or it will initiate new cluster based on appropriateness conformance checker result. If the distance between new event log created by adding this audit trail to cluster event log and existed cluster model greater than or equal the appropriateness percentage defines by user input then this audit trail accepted to join this cluster otherwise it rejected.

Fig. 2. shows a screenshot of constructing miner plug-in. As can be seen, the window for specifying the settings has two parts: first one for setting the parameters used by the heuristics miner and another for setting the parameters of appropriateness conformance checker percentage and maximum depth level to analysis.

The window for showing the results has two parts as shown in Fig. 3. First one for list of observed clusters and for each cluster defines percentage of included audit trails. And the other part shows the corresponding model for selected cluster in first part.

Number of extracted cluster variable based on appropriateness percentage. If we apply clustering with zero percentage we will have only one cluster represent all logs “i.e. in this case there is no clustering”. If we apply clustering with 100% percentage we will have all distinct patterns in event log and from this we can easy define the most common pattern and also less common pattern. Number of clusters increase gradually based appropriateness percentage from zero to 100%.

In our running example we use selected population of size 50 from communication logs in reality mining dataset. The extracted model by heuristics miner plug-in that represent all logs shown in Fig. 4. Result of clustering this event log with zero appropriateness percentage shown in Fig. 5. Result of clustering this event log with 50% appropriateness percentage shown in Fig. 3. Result of clustering this event log with 100% appropriateness percentage shown in Fig. 6.

---

**Input:** whole even log \( L_e \), appropriateness percentage \( P \).

**Output:** list of pairs \( C = \{ (L_i, N_i) \} \), such that \( L_i \) refer to clustered log number \( i \), \( N_i \) refer to model represent clustered log \( L_i \).

**Method:** Perform the following steps:

1. Initialize \( C \) as empty list
2. For each audit trail log \( L \) in \( L_e \) do
   2.1. Let classified := false ; //isClassified flag equal true only if this audit trail \( L \) has been classified
   2.2. For each cluster \( C[i] := (L_i, N_i) \) in \( C \) and classified := false do
      2.2.1. Let \( L' := L \cup \{ L \} \) ; // extend old log with new audit trail
      2.2.2. Let \( P' := \alpha(\lambda(L), N) \) ; //Improved appropriateness metric between old model and new log
      2.2.3. If \( P' \geq P \) then // appropriateness greater than or equal user input appropriateness conformance
         2.2.3.1. Set classified := true ; //this audit trail belong to this cluster
         2.2.3.2. Set \( L := L' \) ; // update old log with new audit trail
         2.2.3.3. Set \( N := \text{heuristics model of } L \) ; // update corresponding model for updated log
   2.2. End if
   2.3. End for
   2.4. If classified = false then
      2.4.1. Let \( L_{new} = \{ \} \) ; // create new log using this audit trail
      2.4.2. Let \( N_{new} = \text{heuristics model of } L_{new} \) ; // corresponding model for new log
      2.4.3. Let \( C_{new} := (L_{new}, N_{new}) \) ; // create new cluster using the new log
      2.4.4. Add \( C_{new} \) to \( C \) ; // add new cluster to clusters list
   2.5. End if
3. End for

**Figure 1.** Algorithm for constructing workflow clusters based on Appropriateness Conformance Checker
Constructing mining plugin

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative-to-best threshold</td>
<td>0.05</td>
</tr>
<tr>
<td>Positive observations</td>
<td>1.0</td>
</tr>
<tr>
<td>Dependency threshold</td>
<td>0.9</td>
</tr>
<tr>
<td>Length-one-loops threshold</td>
<td>0.9</td>
</tr>
<tr>
<td>Length-two-loops threshold</td>
<td>0.9</td>
</tr>
<tr>
<td>Long distance threshold</td>
<td>0.9</td>
</tr>
<tr>
<td>Dependency divisor</td>
<td>1.0</td>
</tr>
<tr>
<td>AND threshold</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- Extra info
- Use all activities connected heuristic
- Use long distance dependency heuristics

Appropriateness Percentage: 50%
Maximum depth level: 3

Figure 2. Constructing miner plug-in

Figure 3. Clustering result using 50% appropriateness percentage

Figure 4. Heuristics miner result for communication event log
5 Related Work

This section shows the other techniques provided to cluster workflow models. In [16] the soundness metric provided which receives a model and a log as input and calculates the percentage of traces that a model can generate and are not in the log. Since the log is assumed to be exhaustive, this metric only works for acyclic models. In [17] use A-priori like approach to extract a relevant feature is a sequence and use k-means as clustering technique. In [18] use sequence clustering based on hidden Markov model, and this methods has been an active field of research especially in connection with challenges in bioinformatics. In [19] try to build structure process by aggregated similar events without clustering the model to several models.

6 Conclusion

In this paper we proposed clustering approach based on appropriateness conformance checker metric. Conformance checker metrics provided to analysis process model and deal with all model attributes and properties. Using appropriateness percentage as input for clustering process easy to understood by user and also easy to use, so we can say clustering using appropriateness percentage expressed the same. The whole approach has been implemented in ProM the process mining framework.

On the other hand the differentiation between modeling language that heuristic miner working with and modeling language that conformance metric working with wasted time. So we will working on unified modeling language between miner and conformance checker.

References: