Sub-process discovery: Opportunities for Process Diagnostics

Raykenler Yzquierdo-Herrera[1], Rogelio Silverio-Castro[1], Manuel Lazo-Cortés[1]

[1]Faculty 3, University of the Informatics Sciences. Habana, Cuba
{ryzquierdo,Silverio,manuelslc}@uci.cu

Abstract. Most business processes in real life are not strictly ruled by the information systems that support them. This behavior is reflected in the traces stored by information systems. It is useful to diagnose in early stages of business process analysis. Process diagnostics is part of the process mining and it encompasses process performance analysis, anomaly detection, and inspection of interesting patterns. The techniques developed in this area have problems to detect sub-processes associated with the analyzed process and framing anomalies and significant patterns in the detected sub-processes. This proposal allows to segment the aligned traces and to form representative groups of sub-processes that compose the process analyzed. The tree of building blocks obtained reflects the hierarchical organization that is established between the sub-processes, considering main execution patterns. The proposal allows greater accuracy in the diagnosis. Based on the findings, implications for theory and practice are discussed.

Keywords: Business process, process diagnostics, process mining, trace alignment.

1 Introduction

Most enterprises and businesses use information systems to manage their business processes [1]. Enterprise Resource Planning systems, Supply Chain Management systems, Customer Relationship Management systems, and systems for Business Process Management themselves are few of the examples that could be mentioned. Information systems register actions in the form of traces as a result of executing instances or cases of a business process. The discovery of processes from the information contained in the traces is part of process mining or workflow mining [2, 3]. The discovery of the process model based on traces allows comparisons with the prescribed or theoretical model. Recent research works describe process mining application as support to the "operationalization" of the enterprise processes. "The idea of process mining is to discover, to monitor, and to improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's information systems" [4].

Most business processes in real life are not strictly ruled by the information systems on the background. This means that although there is a notion of a process,

actors can get away from it, or even ignore it completely. In these environments, it may be wise to start a process improvement or to establish a process quality control to discover the actual running process [5-7].

It is useful to diagnose in early stages of business process analysis. Process diagnostics is part of process mining and it encompasses process performance analysis, anomaly detection, and inspection of interesting patterns [8]. Diagnosis provides a holistic view of the process, the most significant aspects of it and of the techniques that can be useful in further analysis.

The techniques developed in this area have problems to detect sub-processes associated with the analyzed process and framing anomalies, and significant patterns in the detected sub-processes [6, 8].

This proposal allows to segment the aligned traces and to form representative groups of sub-processes that compose the analyzed process. The tree of building blocks obtained reflects the hierarchical organization that is established between the sub-processes, considering main execution patterns. On each case, the building blocks created allows to group segments of the traces which can be significant for analysis.

The rest of this paper is organized as follows: section 2 introduces some related works; in section 3, methodological approach is presented. Furthermore, in section 4, real environment proposed algorithm's application and its results are discussed. Finally, conclusions are given in section 5.

2 Related Works

Among the most used techniques on log visualization, Dotted chart analysis can be found [9]. This technique is a "Gantt charts analogous technique, showing a `helicopter view' of the event log and assisting in process performance analysis by depicting process events in a graphical way, and primarily focuses on the time dimension of events" [8]. Business analysis is manually made from the dotted chart. Manual inspection and comprehension of the dotted chart becomes cumbersome and often infeasible to identify interesting patterns over the use of logs with medium to large number of activities (within few tens to hundreds).

Other commonly used visualization technique is Stream scope visualization. It is based on the event class correlations [10]. Using stream scope visualization, patterns of co-occurring events may be easily recognized by their vicinity. However, the technique is restricted by its unavailability to provide a holistic view of the event log although it visualizes each trace separately.

The use of tandem arrays and maximal repeats to capture recurring patterns within and across the traces is proposed by Bose and Van der Aalst [11]. This work has two limitations, the number of uncovered patterns can be enormous, and the patterns uncovered are atomic (the dependencies/correlations between patterns need to be discovered separately).

The Conformance checking allows to detect inconsistencies/deviations between a process model and its corresponding execution log [12, 13]. Conformance checking as a trend has inherent limitations in its applicability, especially for diagnostic purposes.

It assumes the existence of a preceding process model. However, in reality, process models are either not present or if present are incorrect or outdated [6].

At this point research works that arise with interesting patterns and anomalies detection were shown. Further on, focus will be pointed to sub-processes detection. In this sense, investigations that obtain cluster activity in the analyzed process can be mentioned, which can be useful to understand the context of certain anomalies. These research works are not highly recommended for real environments either is difficult to know the relationship established between the activities that form a group [5, 14]. Those techniques do not provide a holistic view of the process. The Fuzzy Miner discovery technique allows to obtain cluster activities, but it considers that each activity belongs to a single node [15].

The insufficiency for detecting sub-processes makes it complicated, in many occasions, to contextualize detected aspects and to understand its causes. This limitation prevails on the work developed by Van der Aalst and Bose (2012) [8] despite the fact that these authors agree this research yields the best obtained results in diagnostics by making possible to identify recurring patterns and provides a comprehensive holistic view of the process.

3 Methodological approach

Initially, authors present a set of necessary definitions for a better understanding of the proposal.

Definition 1 (Business process): A business process consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations [16].■

Definition 2 (Sub-process): A sub-process is just an encapsulation of business activities that represent a coherent complex logical unit of work. Sub-processes have their own attributes and goals, but they also contribute to achieving the goal of the process. A sub-process is also a process and, an activity, its minimal expression.■

A process can be decomposed into multiple sub-processes using the following workflow patterns:

- Sequence: two sub-processes are arranged sequentially, if one occurs immediately after the other sub-process.
- Choice (XOR or OR): two sub-processes are arranged as options in a decision point; if on each case or process instance only one (XOR) or both in any order (OR) occur.
- Parallelism: two sub-processes are arranged in parallel if both occur simultaneously.
- Loop: A loop occurs when a sub-process is repeated multiple times.

Sub-processes can be decomposed into other sub-processes until the level of atomic activity. This allows building a tree where each level has a lower level of abstraction.

Definition 3 (Trace and event log). Let Σ denote the set of activities. Σ^+ is the set of all non-empty finite sequences of activities from Σ. Any $T \in \Sigma^+$ is a possible trace. An event log $\mathcal{L}$ is a multi-set of traces [8]. ■

Definition 4 (Building block and decomposition into building blocks): Let us denote by *S* the set of all sub-processes that compose the process *P*, $\mathcal{L}$ the event log that represents the executed instances of *P*, $\mathcal{A}$ is the matrix obtained in trace alignment from $\mathcal{L}$, and *Q* is the set of all sub-matrices over $\mathcal{A}$ (Traces Alignment uses the technique developed by Bose and Van der Aalst (2012)[8]).

Let us denote by *Q'* the set of sub-matrices that represent the sub-processes of *S*, such that $Q' \subseteq Q$. Let C^i, C^j, $C^{j+1} \in Q'$, the sequence relationship between two sub-processes represented by C^j and C^{j+1} is denoted by $C^j >' C^{j+1}$. Analogously the choice relationship is denoted by $C^j \#' C^{j+1}$ and the parallelism relationship by $C^j ||' C^{j+1}$. The loop over C^i is denoted by (C^j)*.

Let $s_i \in S$ the process represented by a matrix $C^i \in Q'$ and composed by the sequence of sub-processes represented by $C^j,\ldots, C^{j+k}$ then matrix C^i and the set $\{C^j,\ldots, C^{j+k}\}$ are called *building blocks*. The sub-processes represented by $\{C^j,\ldots,C^{j+k}\}$ are related in one way (sequence, parallelism, OR-XOR or loop).■

General steps of the proposal are presented below.

3.1 Trace alignment

Starting from a workflow log, traces are aligned following the algorithm developed by Bose and Van der Aalst [8]. With the result of aligned traces a file that represents the matrix $\mathcal{A}$ is generated. Trace alignment is a representation of the activities according to a relative order of occurrence and considering cases structure. The order established between activities allows identifying a group of workflow patterns.

3.2 Pre-processing aligned traces

Incomplete cases are determined as cases which do not meet the process end-event. Incomplete cases have gaps ("-") in the columns for the process final activities. These cases can be treated or eliminated; afterwards, traces can be re-aligned. Moreover, trace alignment can be modified in order to assure each column is occupied by a single task.

3.3 Tree of building blocks

The algorithm for determining the tree of building blocks is the following.

```
Algorithm 1. Determining the tree of building blocks
Input: Matrix 𝒜
Output: Tree of building blocks
```

```
1: Create an empty tree
2: Create a building block C^1 and it is associated with
   the root node of the generated tree. C^1 = A
3: if C^i is not a row matrix then
4:  CL = Sequence-Search(C^1). /*CL is a list of obtained
   building blocks*/
5:  if |CL| < 1 then
6:    CL = Loop-Search(C^1)
7:    if |CL| < 1 then
8:      CL = XOR-OR-Search(C^1)
9:      if |CL| < 1 then
10:       CL = Parallelism-Search(C^1)
11:       if |CL| < 1 then
12:         CL = Hidden-Sequence-Search(C^1)
          Endif
        Endif
      Endif
    Endif
13: For each building block i from CL do
14:   i is modified. /* the repeated row and the columns
      that contain only gap from the building block are
      eliminated */
15:   Add i as a child node of root node (C^1)
16:   Apply the Algorithm 1 starting from 3 to i
17:   if tree obtained in the previous step ≠ Ø then
18:     Add to i, as children, the children nodes of the
        root of tree obtained in the step 16
      Endif
    EndFor
   Else
19: Return an empty tree
   Endif
20: Return the generated tree
```

The procedures S*equence-Search*, *Loop-Search*, *XOR-OR-Search*, *Parallelism-Search* and *Hidden*-S*equence-Search* are described below.

S*equence-Search*: The purpose of this proceeding is to determine if the building block (as input) is a process that can be decomposed by a sequence of sub-processes. If the de-composition is possible, it returns a list of detected building blocks, otherwise it returns an empty list. Sequentially ordered sub-processes can be clearly identified. These are separated by one or more activities that appear to occupy an entire column each. Sometimes these activities may not be identified because they could not be mapped in the event log.

Loop-Search: The purpose of this proceeding is to determine if the building block (as input) represents a sub-process repetition. If the de-composition is possible, it returns a list with one building block, otherwise it returns an empty list. To determine

if a building block that represents a sub-process repetition is necessary to identify the initial activity of that specific sub-process. This initial activity can be kept to separate sequences of activities. Those identified sequences constitute the rows of the new building block. Repeated sequences are discarded.

XOR-OR-Search: The purpose of this proceeding is to determine if the building block (as input) is a process that can be decomposed by a choice of sub-processes (OR or XOR). If the de-composition is possible, it returns a list of building blocks detected, otherwise it returns an empty list.

Firstly, authors search the de-composition by XOR. To determine the building blocks that represent options (XOR) in a decision point disjoint sets are constructed with the activities which form the analyzed building block. Originally, there is a set of activities to each building block row; later on, the sets that intersect with some activity are joined. If there is more than one set at the end of this process, then building blocks that represent each of resultant options are created. Otherwise, if there is only one set, then the search to identify the de-composition by OR is performed. In order to do this, base sequences are determined. A base sequence is a row of a building block that is not composed entirely by the join of other rows. Sequences that contain common activities belong to the same set.

Parallelism-Search: The purpose of this proceeding is to determine if the building block (as input) is a process that can be decomposed by a parallelism of sub-processes. If the de-composition is possible, it returns a list of detected building blocks, otherwise it returns an empty list. To determine the building blocks that represent parallel sub-processes, disjoint sets are identified with the activities which form the analyzed building block. Activities belonging to different sets are in parallel, while activities belonging to one specific set are related by other workflow pattern. If more than one set is obtained as result, then the building blocks are formed from these parallel sub-processes.

Hidden-Sequence-Search: The purpose of this proceeding is to determine if the building block (as input) is a process that can be decomposed by a sequence of sub-processes. If the de-composition is possible, it returns a list of detected building blocks, otherwise it returns an empty list.

In this case, it is assumed that the activity or activities which define the sequentially ordered sub-processes are not recorded in the traces. Consequently, possible solutions (de-composition scenarios) are determined considering the issues set out below.

- Each building block that forms a solution can be decomposed by XOR, OR, loop or parallelism.
- The solutions are evaluated and the best are selected, taking into account within the evaluation that formed building blocks decrease the amount of broken loops and parallelisms (e.g. a broken loop is evident when an activity appears multiple times in a row in the analyzed building block; then different instances of the referred activity make appearance on different new building blocks instead a same new building block).

4 Applying the proposal in a real environment and discussion

The technique presented in section 3 has been implemented and the traces of module Management of Roles from National Identification System (SUIN) were analyzed. The SUIN is a system developed by the Cuban Ministry of Interior in conjunction with the Cuban University of Informatics Sciences. The event log did allow determining anomalies in the selected process (31 cases, 804 events, 52 events classes and 3 types of events). The first step was to apply the trace alignment technic developed by Van der Aalst and Bose (2012) [8]. Fig. 1 shows the obtained alignment from the event log.

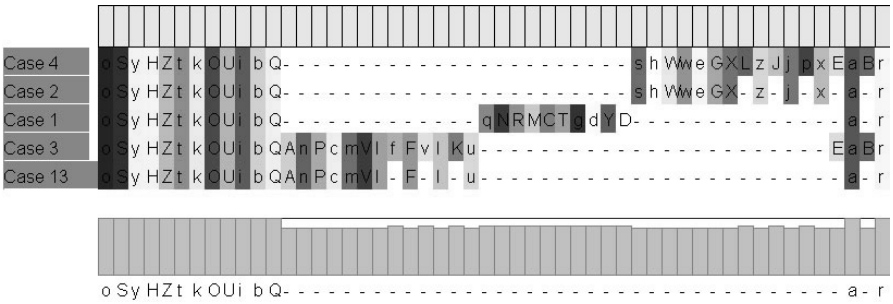

Fig. 1. Trace alignment.

The proposal was applied to obtained a matrix from the alignment (Fig. 1) and afterwards the tree of building blocks, as shown in Fig. 2 (left panel), was obtained.

It can be noticed that: the obtained tree of building blocks may be expanded until all nodes become leaves or until they can no more be decomposed. The edges have different colors to differentiate the used workflow pattern; this is also indicated by a text message in each case (SEQUENCE, XOR, HIDDEN_SEQUENCE).

On Fig. 2 it appears the BB_2_4 building block selected (circle enclosed) which corresponds to the final decomposition resultant sub-process of BB_1_1. The BB_2_4 building block was chosen because it makes it possible to know how the process end. It also contains two cases, the first with frequency of 12 and a second with frequency of 19. This information can be appreciated in the middle table shown on figure 2, which corresponds to each case's occurrence frequency. Neither case's occurrence frequency nor activities' occurrence frequency are used in Algorithm 1, but they are incorporated in the developed tool to make process diagnosis easier.

The first BB_2_4 case is associated to the activity B which represents the event *Roles Management* activity *fault*. It is relevant this process failed 12 of the 31 executed times, representing a 38.7% of faults. Consequently failure causes on the process tested were sought.

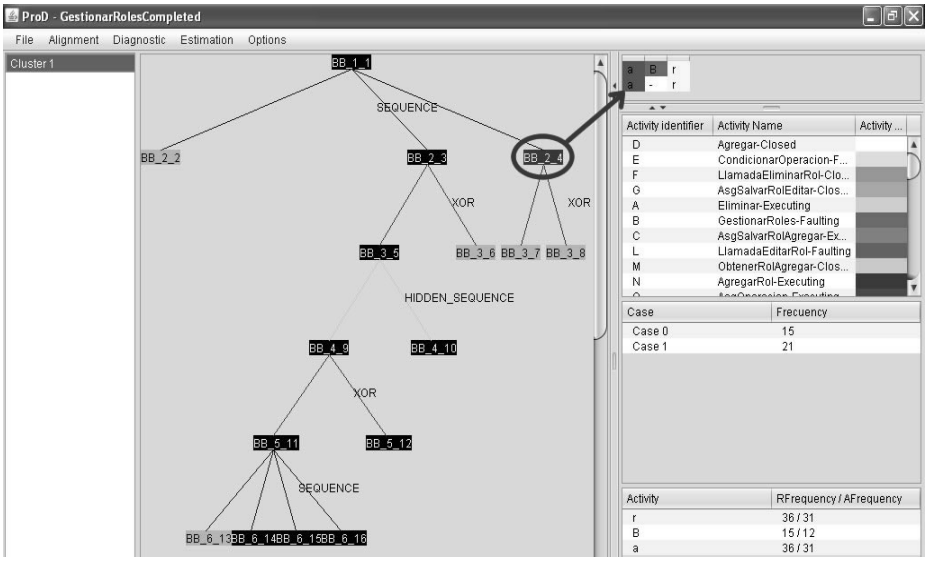

Fig. 2. Process de-composition.

The origin of faults was searched on BB_2_3, which includes possible actions related to add, edit or delete a role. From the BB_2_3 de-composition two building blocks are obtained, the BB_3_5 and BB_3_6, both of them representing choice options. BB_3_6 represents the *Create Role* sub-process and it does not contain any B activity, which indicates that this building block had no influence in the process failure. From the BB_3_5 de-composition two new building blocks are obtained. In which the BB_4_10 represents the *Edit Role* and *Delete Role* sub-processes end-event. In BB_4_10 the event failure appears, it indicates that the failure lays on the sub-processes *Edit Role* and *Delete Role*. Detailed analysis of the Building block BB_4_9 and its de-composition was performed in order to determine the sequence of activities that led to failures in the *Edit Role* sub-process (represented by BB_5_11) and Delete Role sub-process (represented by BB_5_12). This detected sequence of activities is useful for future warnings prior to the possibility of a failure. Authors were able to identified specific cases in which failures occurred. Knowing the cases and events where the failure took place, the involved users were identified.

The technique developed by the authors, as well as the technique developed by Van der Aalst and Bose (2012) [8] allows detecting interesting patterns and provides a holistic view of the process. The proposal also allows detection of sub-processes that compose the analyzed process. The detected sub-processes enclose anomalies and interesting patterns, something that is not satisfied by the techniques discussed in section 2.

Another advantage of the present research is that it combines the cases and activities occurrence frequency analysis with the staged analysis from correctly structured sequence events on sub-processes. This contributes to the understanding of the failure causes and therefore a subsequent possible process improvement.

An important contribution of this work is that the anomalies detected can be framed in a context. For example, the detected anomalies in the analyzed process are located in sub-processes Edit Role and Delete Role.

The developed tool was also applied to analyze the process "Check Management" in the bar Gulf View and the restaurant Aguiar, both places belonging to the National Hotel (Cuba). Main characteristics of the process for both event logs, which supported the auditing of the process, were identified [17].

5 Conclusion

Process diagnostics can be useful for detecting patterns and anomalies in the analyzed process. The techniques developed in this area have problems to detect sub-processes associated with the analyzed process and framing those anomalies and significant patterns in the detected sub-processes.

This proposal allows to segment the aligned traces and to form representative groups of sub-processes that compose the analyzed process. The obtained tree of building blocks reflects the hierarchical organization that is established between the sub-processes, considering main execution patterns. On each case, the building blocks created allow to group segments of the traces which can be significant for analysis.

The proposal allows detecting interesting patterns and provides a holistic view of the process. Another advantage of the present research is that the interesting patterns detected can be framed in a context. The discovery of sub-processes that compose the analyzed process, its dependencies and correlations allow greater accuracy in the diagnosis. All this is possible thanks to the combination of the cases and activities occurrence frequency analysis with the staged analysis from correctly structured sequence events on sub-processes.

6 References

1. Hendricks, K.B., Singhal, V.R., and Stratman, J.K.: The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations. Journal of Operations Management, vol. 25, issue 1, pp. 65--82 (2007)
2. Agrawal, R., Gunopulos, D., and Leymann F.: Mining Process Models from Workflow Logs. In EDBT '98 Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology. Springer-Verlag London, UK (1998)
3. Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. ACM Transactions on Software Engineering and Methodology, pp. 215--249 (1998)
4. Aalst, W.M.P.v.d.:Process Mining. Discovery, Conformance and Enhancement of Business Processes. Springer Heidelberg Dordrecht London New York (2011)
5. Dongen, B.F., Adriansyah, A.: Process Mining: Fuzzy Clustering and Performance Visualization. In Business Process Management Workshops, S. Rinderle-Ma, S. Sadiq, and F. Leymann, Editors, Springer Berlin Heidelberg. pp. 158--169 (2010)

6. Bose, R.P.J.C., Aalst, W.M.P.v.d.: Trace Alignment in Process Mining: Opportunities for Process Diagnostics. In International Conference on Business Process Management (BPM'2010), Springer-Verlag Berlin, Heidelberg (2010)
7. Song, M., Günther, C.W., Aalst, W.M.P.: Trace Clustering in Process Mining. In Business Process Management Workshops, Milano, Italy, Lecture Notes (2009)
8. Bose, R.P.J.C., Aalst, W.M.P.v.d.: Process diagnostics using trace alignment: Opportunities, issues, and challenges. Inf. Syst., vol. 37, issue 2, pp. 117--141 (2012)
9. Song, M., Aalst, W.M.P.V.d.: Supporting process mining by showing events at a glance. In 17th Annual Workshop on Information Technologies and Systems (WITS) (2007)
10. Günther, C.W.: Process Mining in Flexible Environments, Eindhoven University of Technology, Eindhoven (2009)
11. Bose, R.P.J.C., Aalst, W.M.P.v.d.: Abstractions in Process Mining: A Taxonomy of Patterns. U. Dayal, et al., Editors, Springer Berlin / Heidelberg. pp. 159--175 (2009)
12. Rozinat, A., Aalst, W.M.P.v.d.: Conformance checking of processes based on monitoring real behavior. Inf. Syst., vol. 33, issue 1, pp. 64--95 (2008)
13. Adriansyah, A., Dongen, B.F.v., Aalst, W.M.P.v.d.: Towards Robust Conformance Checking. In BPM 2010 Workshops, Proceedings of the 6th Workshop on Business Process Intelligence. Lecture Notes in Business Information Processing. Springer, Berlin (2011)
14. Aalst, W.M.P.V.D., Rubin, V., Verbeek, H.M.W., Dongen, B.F.V., Kindler, E., Günther, C.W.: ProcessMining: A Two-Step Approach to Balance Between Underfitting and Overfitting. Software and Systems Modeling, vol. 9, issue 1, pp. 87--111 (2009)
15. Günther, C.W., Aalst, W.M.P.v.d.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-Perspective Metrics. in International Conference on Business Process Management (BPM 2007). Lecture Notes in Computer Science. Springer, Berlin (2007)
16. Weske, M.: Business Process Management. Concepts, Languages, Architectures, ed. S.-V.B. Heidelberg (2007)
17. González, L., Suárez, M.: Procedure for the application of process mining techniques in auditing processes [Degree]. Faculty of Industrial Engineering, Polytechnic Institute José Antonio Echeverría, Havana, Cuba (2012)