Detecting anomalies in the organizational perspective using Process Mining

Alejandro Fuentes-delaHoz and Dr. Marcos Sepúlveda

Computer Science Department
School of Engineering
Pontificia Universidad Católica de Chile
afuentesd@uc.cl, marcos.sepulveda@ing.puc.cl

Summary. The increasing popularity of Process Aware Information Systems (PAIS) has produced new challenges to the auditing of processes supported by this type of systems. Some methods have been proposed for anomaly detection based in Process Mining, but they only analyze data from the control-flow perspective of business processes. This paper explores the organizational perspective and proposes a new anomaly detection method that detects three new types of anomalies that are originated in the organizational perspective. A test case is presented to show that is possible to detect these new types of anomalies, and that the whole anomaly detection process gets improved.

Key words: process mining, anomaly detection, organizational perspective

1 Introduction

Process Aware Information Systems (PAIS) have become a key component of the IT infrastructure of a large amount of organizations worldwide [1]. As more information is managed and more processes are supported by this type of systems, there is an increase need for improved auditing and internal controls [2] of those processes. This new challenges also emerge from new industry and compliance requirements (e.g., Sarbanes-Oxley [3]).

In this context, Process Mining emerges as a research discipline that could provide a powerful group of tools for the analysis of business processes, like process discovery [4], conformance checking [5] and anomaly detection [6].

Some anomaly detection methods, based in process mining techniques, have been proposed for PAIS ([6] and [7]). These algorithms allow the detection of anomalies only in the control-flow perspective of business processes. However, excluding the organizational and data perspectives from the analysis, particularly in anomaly detection, could lead to miss relevant anomalies.

This paper proposes a new anomaly detection method that detects three new types of anomalies that are originated in the organizational perspective.

The remainder of this paper is organized as follows. Section 2 presents some related work in the area of anomaly detection and process mining. Section 3

explains the proposed model and the main differences with the current model. Section 4 shows a complete example and the comparison between the current and the proposed model, using the process mining toolkit ProM [8]. Section 5 presents the conclusions of this work and the future work .

2 Related Work

Process mining is a research discipline [9], which proposes a group of techniques to analyze event logs from business process in order to obtain different types of useful information about the process. Most Process Mining algorithms analyze only the control-flow perspective, such as process discovery [4] or conformance checking [5].

However, some techniques have been proposed that analyze the organizational and data perspective. In the first case, these techniques are known as "organizational mining" and aim to obtain an organization model of the business process. Two of the main papers in this area are [10] and [11]. In [11], four methods are proposed to analyze the organizational data, three of them focus in the organizational roles and the latter focuses in the group of performers (teams) that can be detected in the event log.

Other process mining techniques analyze the data perspective, e.g., decision mining[12]. In this case, the algorithms usually analyze both the control-flow and the data perspective at the same time.

Anomaly detection in PAIS is also a research field where some process mining techniques have been proposed. Two main approaches exist. In [6], a method is proposed focused on the control-flow perspective, which consists of two stages: first, a process discovery step using the alpha algorithm [6] is performed for creating a model; then, they apply a conformance checking algorithm [5], and use the fitness metric as the anomaly scoring.

The method proposed in [7] is more complex but similar to the previous model. In this method, the process discovery step produces several process models, and not only one as in the previous approach. Then, they add a new step to evaluate the discovered models and select the model with the highest fitness and appropriateness [5]. Then a conformance checking algorithm is used with the selected model, splitting the log in normal instances (those that fit in the selected model), and anomaly instances (those that do not).

Both methods are able to detect anomalies in the control-flow perspective, but, as the same authors said, it is possible that the anomalies exist in the other perspectives, and it is important to consider them in the anomaly detection process. Also it is interesting to consider the example of the decision mining algorithm, which analyzes two perspectives at the same time. Therefore, we have considered feasible to analyze the control-flow and the organizational perspectives at the same time, in order to detect anomalies that could appear in any of these two perspectives.

3 Detecting anomalies originated in the organizational perspective

The proposed technique aims to detect three types of anomalies that could be found in events logs from PAIS and that are originated in the organizational perspective.

These types of organizational anomalies are related with groups of performers (also known as teams), that work together in a subset of process instances. Hence, the first step of this technique consists in obtaining the teams that work together in a subset of process instances. This is achieved by using part of the Organizational Mining algorithms proposed in [11]; in this case, the "Working Together" algorithm (Fig. 1).

Fig. 1. Group of performers that work together (also known as teams)

After obtaining the teams, the process instances are grouped by the team that performs the activities (Fig. Fig. 2). As a result, there are as many subsets of process instances as teams in the event log.

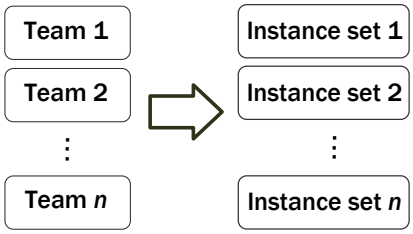

Fig. 2. Subsets of process instances

Having process instances grouped by teams, it's now possible to start looking for anomalies in the instances.

3.1 Teams with a small number of instances

A team with a small number of process instances in comparison with other teams could be considered as an anomaly. This could be related with executions made

in some kind of emergency, out-of-schedule, or by a group of people that do not often perform this process together. These subsets of process instances are marked as potential anomalies, to allow further analysis and to audit the possible causes of their occurrence.

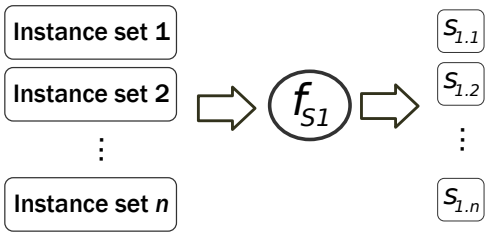

Fig. 3. Teams with a small number of process instances

To detect this kind of anomalies, a parameter p must be defined, that delineates the minimum number of instances (as a percentage of the total amount of process instances) that a team must have. This value is defined for each process and could be based on historical or statistical data.

The comparison of the number of instances per team and this parameter p, defines the first scoring of the proposed technique $S_{1.i}$ (Fig. 3). The function f_{S1} used to obtain this first scoring, is shown in Equation 1.

$$f_{S1}(x) = \begin{cases} 0 \text{ if } x > p \\ 1 \text{ if } x \leq p \end{cases} \tag{1}$$

3.2 Anomalies within a subset of process instances associate with a team

Current anomaly detection methods analyze simultaneously all the process instances available in the event log, without grouping the instances in any way. This could cause that some situations that could be considered as anomalies are not noticed. By applying the current anomaly detection methods separately to the subset of process instances associated to each team, allows detecting a new type of anomaly.

Table 1 shows an example of this kind of anomaly. The patterns ABCDE and ABCD are normal in the event log as a whole. However, the pattern ABCD is mainly performed by the team 3, so the instance #40 (performed by the Team 2) could be considered an anomaly within its subset of process instances. Therefore, by applying the algorithm to the process instances grouped by teams, the instance #40 will be marked as a potential anomaly.

To detect this kind of anomaly, the proposed method uses the same techniques introduced in [6] for finding anomalies in the control-flow perspective. In this case, applied to a subset of process instances that share the same team of

Instance #	Team	Activities	Global Anomaly	Local Anomaly
1	Team 1	ABCDE	No	No
...	...	...	...	...
19	Team 1	ABCDE	No	No
20	Team 1	ABCDE	No	No
21	Team 2	ABCDE	No	No
...	...	...	...	...
39	Team 2	ABCDE	No	No
40	Team 2	ABCD	Yes	Yes
41	Team 3	ABCD	Yes	No
...	...	...	...	...
59	Team 3	ABCD	Yes	No
60	Team 3	ABCD	Yes	No

Table 1. Example of a local anomaly. Instance #40 is an anomaly within its subset of process instances

performers, allows detecting control-flow anomalies within each subset, as shown in Fig. 4. The result of this method is a second scoring for each process instancei, labeled as $S_{2.i}$.

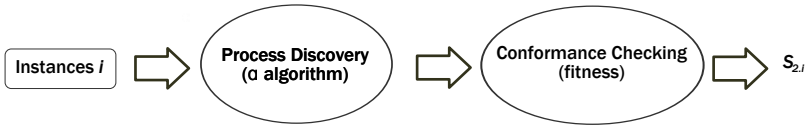

Fig. 4. Local anomaly detection

3.3 Team anomaly

With process instances grouped by team, it's also possible to detect singular behaviors on the process instances of a given subset in comparison to the process instances of other subsets, which could also be considered as an anomaly. This behavior could have several causes, but it's anyway relevant to detect it.

Table 2 shows an example of this kind of behavior. In this case, the subset of process instances associate with the Team 3 or the Team 3 itself could be considered as an anomaly because all the process instances within this subset have a different behavior in comparison to the rest of the instances of the whole event log. This is also true, even considering that exist instances like the instance #20 (performed by the Team 1) that show the same behavior of the instances performed by the Team 3.

In this case, the first step in the proposed technique consists of applying process discovery to each subset of process instances grouped by team. This

Instance #	Team	Activities	Global Anomaly	Team anomaly
1	Team 1	ABCDE	No	No
...	...	...	...	...
19	Team 1	ABCDE	No	No
20	Team 1	ABCD	Yes	No
21	Team 2	ABCDE	No	No
...	...	...	...	...
39	Team 2	ABCDE	No	No
40	Team 2	ABCDE	No	No
41	Team 3	ABCD	Yes	Yes
...	...	...	...	...
59	Team 3	ABCD	Yes	Yes
60	Team 3	ABCD	Yes	Yes

Table 2. Example of team anomaly. Process instances performed by Team 3 are different to most of the other process instances

produces as many models as teams exist in the event log. The second step is a conformance checking between each of the models discovered in the previous step and every subset of process instances, as shown in Table 3 .

	$Model_1$	$Model_2$	...	$Model_j$	*Team Scoring*
Team 1	$f_{1,1}$	$f_{1,2}$	...	$f_{1,j}$	$S_{3.1}$
Team 2	$f_{2,1}$	$f_{2,2}$	...	$f_{2,j}$	$S_{3.2}$
...	...	...	...	...	...
Team i	$f_{i,1}$	$f_{i,2}$	...	$f_{i,j}$	$S_{3.i}$

Table 3. Team anomaly scoring matrix

With this values it's possible to obtain a scoring $f_{i,j}$ for each team i and model j, using the "token game" approach of conformance checking algorithm [5]. Then, for each team i, a final scoring $S_{3.i}$ is calculated for all the process instances that are part of the subset associated to team i, as shown in Eq. 2.

$$S_{3.i} = \frac{\sum_{i \neq j}^{n} f_{i,j}}{n-1} \tag{2}$$

3.4 Analysis of results

With the procedure previously described, a three-dimensional vector $v_{S.i} = (S_{1.i}, S_{2.i}, S_{3.i})$ is obtained for each process instance i. Each value reflects the level of anomaly presented by the process instance on the three new anomaly types.

These values could be used separately to identify potential anomalies or they could be used to obtain an overall anomaly scoring for a given process instance $S_{f.i}$, through the following equation (Eq. 3):

$$S_{f.i} = aS_{1.i} + bS_{2.i} + cS_{3.i} \tag{3}$$

The weights (a, b and c) on Ec. 3 allows giving more or less importance to the different anomaly types, according to the needs or emphasis of the auditing process on a given organization.

This final scoring ($S_{f.i}$), will in turn establish a ranking of process instances, by sorting from the highest to the lowest final scoring. This ranking is relevant because it singles out the process instances that display the greatest amount of anomaly. Any auditing or reviewing process should give a higher priority to the top process instance in order to analyze them in more detail.

4 Test case based in ProM

To test the result of the proposed method, four event logs were built, with different number and types of anomalies. Both the current and the proposed method were applied to these event logs. A comparison of them is presented, including the ability to detect the new types of anomalies that were included in the event logs.

These tests were made using several plugins from the ProM [8] tool.

4.1 Event log building

The first step in this test case, consist in building four event logs, with different number and types of anomalies as is shown in Table 4.

Event Log Name	# of Teams	Low number anomaly	Local Anomalies	Team Anomalies
Event Log A	3	No	Yes, vary with each team	No
Event Log B	4	Yes, team 2	Yes, vary with each team	No
Event Log C	4	No	Yes, vary with each team	Yes, team 3
Event Log D	4	Yes, team 1	Yes, vary with each team	Yes, team 3

Table 4. Events logs for the tests

4.2 Obtain executors groups from an event logs

The ProM Organizational Mining plugin [8] was used to obtain the groups of performers (teams) that exist in each event log. In this case, the "Working Together" algorithm was used [11].

After the teams were obtained, the next step is to group the process instances by team. As a result of this step, each event log is divided in as many subsets of process instances as the number of teams in the event log.

4.3 Scoring with the current model

In this step, the anomaly detection method proposed in [6] is applied to the four test event logs. This algorithm has two steps: first a process discovery algorithm is applied to each event log. In this particular case, the alpha algorithm [4] is used. After that, a conformance checking algorithm is used to compare the discovered process model with the real data in the event log. In this case, the "token game" approach is used for the comparison [6].

This model produces a scoring value (between 0 and 1) that measures the level of anomaly of each process instance. As mention before, this algorithm only analyzes the control-flow perspective, and therefore, it is able only to find anomalies that occur in this perspective.

The result of the analysis with the current method is shown in Figures 5, 7, 9 and 11.

4.4 Scoring with the proposed model

In this step, the proposed method is applied to the four test event logs. Three scores are obtained, that are then used to calculate a final scoring per process instance.

The result of the analysis with the proposed method is shown in Figures 6, 8, 10 y 12.

4.5 Comparison of results

Following, the results of the two methods will be compared. The proposed method was compared against the current one, in two ways: first, comparing the partial scorings produced by the proposed method; and then, comparing the final scoring and ranking.

Partial Scorings

Each event log contains additional information (a flag) that identifies the process instances that contain any of the three types of anomalies. This extra information was used to evaluate the result of the proposed method. By comparing the partial scorings against the flags, we were able to show that each scoring could detect a particular type of anomaly.

In particular, the algorithm to detect a low number of process instances was able to detect that the team #2 from the event log B and the team #1 from the event log D had an lower percentage of instances than the parameter p (5% in this case). The proposed method was also able to detect that the team #3 of the event log C was an anomaly team, with a higher number of anomalies compared with the other teams.

Total Scoring and Ranking

In this section the rest of the results of this paper are presented. The charts in this section show the scoring (in blue) and the total number of anomalies (in red) per each instance, in descendant order by scoring.

– **Event log A (Fig 5 & 6):** this event log contained neither low number of instances anomalies nor team anomalies. The current method (Fig.5) and the proposed method (Fig.6) present a similar behavior and detect the same amount of anomalies.

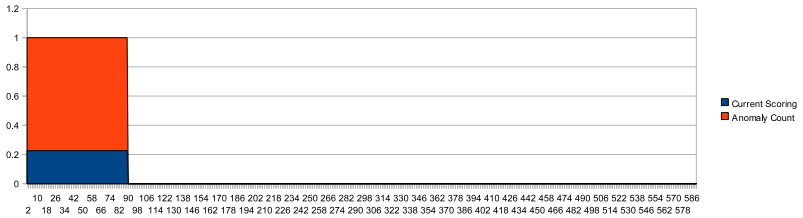

Fig. 5. Case A: Current Scoring

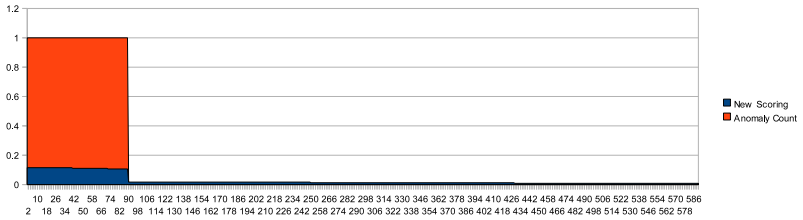

Fig. 6. Case A: New Scoring

– **Event log B (Fig.7 & 8):** this event log contains a team with a low number of instances. The current method (Fig.7) is not able to detect this type of anomaly. This explains the part of the chart that has anomalies (red zone), but whose scoring is zero. The proposed method (Fig. 8) is able to detect this type of anomalies. Therefore, any instance with an anomaly has a score greater than zero.
– **Event log C (Fig.9 & 10):** in this case, the current method (Fig. 9) could not detect the team anomaly of team #3. This explains that in the chart some instances with more anomalies obtain a lower scoring than other instances with fewer anomalies. The proposed method (Fig. 10) is able to detect the anomalies in the process instances performed by team #3. In this case, the ranking is a good predictor of the level of anomaly of any instance.

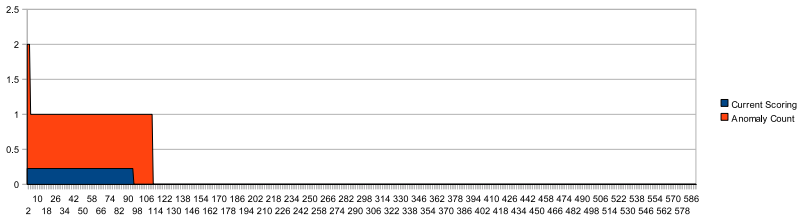

Fig. 7. Case B: Current Scoring

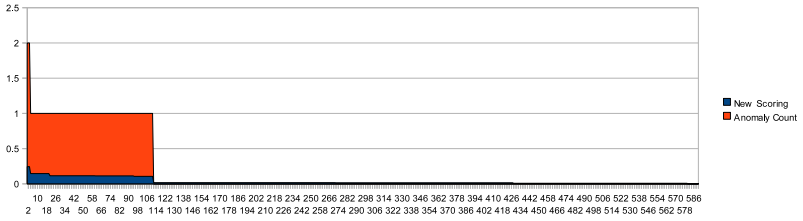

Fig. 8. Case B: New Scoring

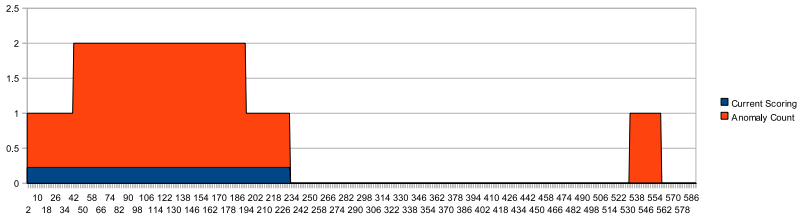

Fig. 9. Case C: Current Scoring

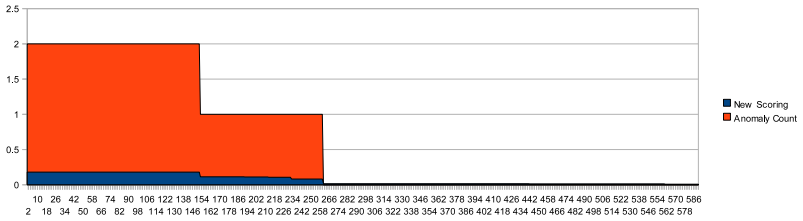

Fig. 10. Case C: New Scoring

– **Event log D (Fig. 11 & 12):** this event log contains the three types of anomalies presented in the previous sections. The current method (Fig 11) is able to detect only some of the anomalies in the process instances. This explains the instances with anomalies (in red) that appear at the right of the chart. Moreover, the ranking is not able to properly score all process instances. The proposed method (Fig. 12) is able to identify the three types of anomalies. In addition, the ranking obtained reflects the number of anomalies in the process instances, sorting them from the ones with more anomalies to the instances with fewer or no anomalies.

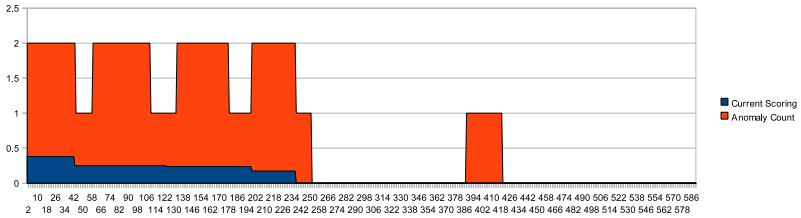

Fig. 11. Case D: Current Scoring

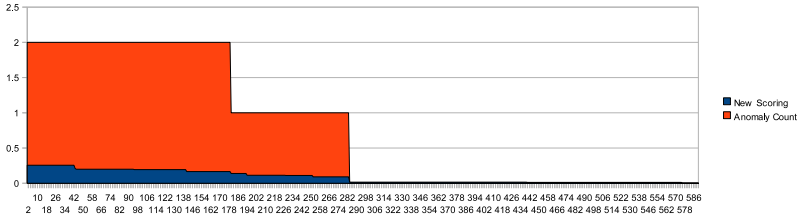

Fig. 12. Case D: New Scoring

5 Conclusions and future work

This paper presents a new method for anomaly detection on PAIS. This is a novel approach that takes into account the organizational perspective in the anomaly detection process, in comparison with the previous algorithms that only focused in the control-flow perspective. The method detects three new types of anomalies that are originated in the organizational perspective.

An event log with these three types of anomalies related with the organizational perspective was built and used to verify the proposed method. We were able to detect the introduced anomalies and produced a better anomaly scoring

than the current method. This is very relevant because it could lead to a better auditory of business processes, where process instances with a larger number of anomalies obtain a higher scoring, and therefore should be considered for further analysis.

This work is still in an initial stage, and a lot of work needs to be done. First, we plan to address other types of anomalies, those related with the roles defined in a process. We think our approach could also be applied to discover these types of anomalies using organizational mining techniques.

The proposed method should also be validated on real business processes, to prove that this type of organizational-related anomalies is relevant in real situations.

References

1. Dumas, M., van der Aalst, W., Ter Hofstede, A.: Process-aware information systems: bridging people and software through process technology. Wiley-Blackwell (2005)
2. van der Aalst, W., van Hee, K., van Werf, J., Verdonk, M.: Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. Computer (2010) 90–93
3. P., S., M., O.: Public Law No. 107-204. Washington, DC: Government Printing Office (2002)
4. Van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering (2004) 1128–1142
5. Rozinat, A., van der Aalst, W.: Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: Business Process Management Workshops, Springer (2005) 163–176
6. van der Aalst, W., de Medeiros, A.: Process mining and security: Detecting anomalous process executions and checking process conformance. Electronic Notes in Theoretical Computer Science **121** (2005) 3–21
7. Bezerra, F., Wainer, J., van der Aalst, W.: Anomaly Detection using Process Mining. Enterprise, Business-Process and Information Systems Modeling (2009) 149–161
8. van Dongen, B., de Medeiros, A., Verbeek, H., Weijters, A., van der Aalst, W.: The ProM framework: A new era in process mining tool support. Applications and Theory of Petri Nets 2005 (2005) 444–454
9. Van der Aalst, W., Weijters, A.: Process mining: a research agenda. Computers in Industry **53**(3) (2004) 231–244
10. Van der Aalst, W., Song, M.: Mining Social Networks: Uncovering interaction patterns in business processes. Business Process Management (2004) 244–260
11. Song, M., van der Aalst, W.: Towards comprehensive support for organizational mining. Decision Support Systems **46**(1) (2008) 300–317
12. Rozinat, A., van der Aalst, W.: Decision mining in ProM. Business Process Management (2006) 420–425