WAM! - WorkAround Mining in Healthcare

Wouter van der Waal

Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, the Netherlands

Keywords
Workarounds, Automated Detection, Event data, Healthcare, Process Mining, Business Process Analysis

1. Problem Statement

Many organizations use standard operating procedures to streamline their work. When procedures are clear, people know what to do. Still, it often happens that work is performed in a way that is different from the prescribed procedure. When confronted with unexpected situations, limited time, or a lack of resources, workers may be unable to follow a procedure and may feel compelled to perform a workaround to solve a problem [1].

Some workarounds are beneficial and can be leveraged to improve organizational procedures [2, 3]. In other cases, not following a procedure may be harmful or outright dangerous. Workarounds can result in noncompliance, privacy issues, or negative effects in the process downstream [4, 5]. Whatever the effect, it is important that process owners are provided with insights into the occurrence of workarounds. These insights can help to prevent workarounds from happening again or to improve the concerned procedure [3]. In addition, being able to structurally and comprehensively identify workarounds would allow for the monitoring of their emergence, diffusion, and evolution, potentially enabling process analysts to detect and respond to new workarounds faster than with current techniques.

To date, most studies in which workarounds were identified and analyzed relied on qualitative methods, primarily through interviewing and observing users during their work [3]. This approach has led to valuable insights related to the mechanics and effects of workarounds, as well as the motivations of the people using them [6, 4, 7]. However, the use of qualitative methods is labor-intensive; furthermore, users may not disclose their normal behavior when they are aware of being observed [8]. To address these drawbacks, my focus is on the use of event logs, process mining, and other quantitative analysis techniques. The state of the art shows that workarounds can be detected and monitored with this data [9]. However, it is unclear to what extent this information can be used to discover new workarounds when starting from the data. Note that if both the workaround and the normative process occur outside of the system, there will not be any information available to detect this. While these workarounds may be discovered using interviews, they cannot be found using event logs.

BPM'22: International Conference on Business Process Management, September 11–16, 2022, Münster, Germany
† Dissertation Supervisors: Hajo A. Reijers & Inge van de Weerd
w.g.vanderwaal@uu.nl (W. v. d. Waal)
0000-0001-6514-7570 (W. v. d. Waal)

Earlier work shows that event logs from a Health Information System (HIS) can be used to automatically detect and monitor known workarounds [9]. I stay in line with this focus on the healthcare domain. Studies have shown that in hospital settings specifically workarounds are a widespread phenomenon [10]: Nurses share each other's passwords to save time, physicians send each other X-rays via WhatsApp to get quick second opinions, and secretaries use shadow systems on paper to track the department's occupation. The principles behind my work are nonetheless transparent and I expect that many of the patterns can be transferred to other domains.

This leads to the following research question:

How can we use event data stored in HISs to detect workarounds?

2. Related Work

As illustrated in Figure 1, workarounds occur when users intend to reach a goal but perceive a block to do so using the official procedure [1]. Similar to the conformance-checking field of process mining, we can look for differences between process variants to detect them [11]. However, we see two important differences. First of all, a workaround requires an *intent* to reach a business goal. So, fraud, deception, and errors are not in scope. Secondly, the user is unable to achieve this goal by using the intended procedure [1]. So, accidentally following a different route is not a workaround.

In addition, while non-conformance usually supposes strict rules [12, 13] or a known process model to conform to [14], workarounds can occur without these. Deviance mining also uses process mining but looks at smaller deviations between processes [15], which may also be useful to mine for workarounds. Quite different from a deviance, a workaround can be very common. If a workaround is sufficiently effective, it may be shared throughout the organization, potentially becoming more common than the official procedure [16].

One approach specifically used to automatically detect workarounds is to use deep learning [8]. Neural networks are trained to recognize different workaround types from event logs. These methods can be difficult to use in practice because they require a large amount of labeled training

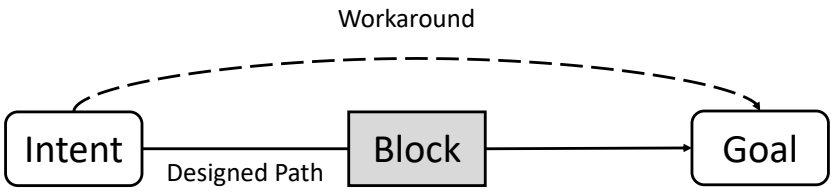

Figure 1: An illustration of the definition of a workaround. A worker has an intent to reach a goal, but is prevented to do so by a block on the designed path. The worker can then deviate from the path to work around that block to reach the goal.

and testing data. In addition, even if neural networks reach a high classification accuracy, it is difficult to explain why this is happening [17], which is often required if you want to use the results in a healthcare environment.

Alternatively, Outmazgin and Soffer define six workaround patterns, four of which can be used to detect workarounds in event logs [18]. These patterns require very strict definitions of what is considered a workaround, such as exact thresholds for activity duration or a full process model. While these patterns can be used to detect known workarounds, we cannot discover new workarounds with them.

Outside of the control-flow, the time, resource, and data perspectives are valuable for conformance checking [11]. By investigating workarounds discovered using qualitative methods, such as interviews and observations, previous studies show that these perspectives can also be used to recognize different types of workarounds [9]. We will continue this multi-perspective approach to investigate if, in addition to recognizing workarounds, we can discover new workarounds using event logs.

3. Intended Approach

As described in the previous section, detecting workarounds has similarities with several approaches in process mining, such as conformance checking. Both focus on differences between the intended and actual process. I will investigate which differences that are used in process mining fields other than workaround mining can also be used to discover workarounds.

Many workarounds were discovered through interviews in previous studies in five different healthcare organizations [9]; a general hospital, two district hospitals, and two specialized centers. They are documented as 'workaround snapshots' and include a description of (1) the setting in which the workaround was found, (2) the workaround compared to the normative process, (3) a motivation, and (4) the expected effect of the workaround on cost, time, quality, and flexibility. Using the comparison between the workaround and the normative process, I can determine if and how each workaround could be monitored using (event log) data, resulting in more patterns in addition to the ones extracted from literature.

While the detection of potential workarounds can be performed in a highly automated fashion, the actual confirmation of the occurrence of workarounds still needs to be done by domain experts. In other words, while the defined patterns can show potentially interesting differences between process variants, only domain experts can determine if a difference is a workaround.

To test the patterns and techniques for workaround mining, I intend to use case studies. Depending on the research question, I can extract data from different wards in a hospital from the HIS over select periods. This data can then be transformed into event logs in which I look for the patterns. Finally, I further evaluate results that indicate a workaround in a semi-structured interview with an expert of the corresponding ward or process. The expert can determine if the observed behavior is normative, a workaround, or something else.

4. Preliminary Findings

By investigating the literature and snapshots, I created a Semi-automated WORkaround Detection (SWORD) framework containing 22 patterns [19]. Table 1 shows an overview of the patterns. I created this framework in the two steps as described in the previous section: 1) a literature study to determine how process mining topics other than workarounds mining define differences between process instances and 2) finding differences between the workarounds and normative processes discovered in earlier studies. For more details on this method, I refer to the full paper.

Note that the framework is extensible, meaning that new patterns can be added. In addition, while the focus of the framework is on workarounds, many of the patterns can be applied to other process mining fields that focus on differences between process variants.

In the same study, I tested three of the patterns on real hospital data. While the results were not verified by experts, they were nonetheless promising; I discovered the occurrence of logging measurements in batches (a previously known workaround), unexpected differences in the delay between taking measurements, and their logging for different types of measurements.

In my current (unpublished) work, I have applied the patterns to different wards in a hospital and held interviews with process experts about the observed outliers. The experts were able to verify multiple outliers as being workarounds, including previously unknown workarounds.

For example, in the emergency room (ER), a physician needs to place a mark in the file if (s)he sees a patient for the first time. This activity can be seen in event logs as 'Seen by physician'. The 'Delay between the start of trace and activity is out of bounds' pattern shows that there are a few cases where this happens a few days after admission to the ER. The expert could explain this behavior as a workaround. If a physician does not perform the required task, administrative personnel update the logs to make sure they reflect that the patient was seen by a physician.

5. Open points

5.1. Better Detection

There are two major points to improve workarounds detection using the patterns.

Firstly, how do we apply the patterns? Currently, I intentionally keep this simple by looking for the largest deviation(s) from the mean. For example, in the confirmed workaround in the previous section, I looked for the events that occurred the most standard deviations away from their meantime. This approach is obviously quite simple and could be improved with more advanced statistics.

Secondly, we have seen in previous work that many workarounds can be detected with multiple patterns. The example workaround was detected by looking for an activity that occurs at an unusual moment, but we can also see that the same trace takes longer than usual. If we combine the information of multiple patterns, this could also improve the workaround detection.

While both points may enhance the quality of workaround detection, it is difficult to exactly quantify this improvement. The usual approach for this is to compare the results to established labels in an extensive dataset using measures such as precision and recall. However, this is not possible for workaround detection, since there is no ground truth available. While experts can

manually determine if a single trace is a workaround or normative, it is unfeasible to do so for large event logs containing thousands of traces.

Regardless, it is important to determine if and to what extent potential improvements enhance the detection rate. It may be sufficient to measure this in a smaller event log. In addition, instead of having an experienced high-level expert evaluate traces, the (initial) evaluation may be performed by a less-involved expert, such as a medical student. Both options would decrease the quality of the comparison, but they make the evaluation more feasible to perform.

5.2. Future Work

Being able to detect more workarounds using event logs allows us to further investigate them. From a sociological point of view, there are many questions about where, when, and how workarounds emerge and evolve. If we can monitor them using event logs, we can give more insight into this. We may potentially even pinpoint when and where a workaround begins and see how it spreads through the organization.

We can also take a more practical approach and investigate the effects of workarounds. If we can reliably distinguish between the normative process and a workaround (or different types of workarounds), we can link this to other available data, such as costs, quality of care, or patient satisfaction using statistical measures such as regression. This would allow organizations to better decide what to do with specific workarounds.

A. Appendix

Table 1: Workaround detection patterns

	Detection pattern	Explanation
Control-flow	Occurrence of an activity	A specific activity occurs
	Occurrence of recurrent activity sequence	A recurrent activity sequence occurs within a trace
	Frequent occurrence of activity	An activity frequently occurs within a trace
	Occurrence of activities in an order different from process model	The order of activities in a trace is other than in a predefined process model
	Occurrence of mutually exclusive activities	Specific activities occur that are mutually exclusive within a trace
	Occurrence of unusual neighboring activities	An activity is directly followed by an activity other than usual
	Occurrence of directly repeating activity	An activity is immediately repeated within a trace
	Missing occurrence of activity	A specific activity is missing in the trace

Continued on next page

Table 1: Workaround detection patterns (Continued)

	Detection pattern	Explanation
Data	Data object with value outside boundary	The value of a data object deviates from the usual values
	Change in value between events	Data values change unexpectedly between events
	Specific information in free-text fields	Information is logged in free-text fields instead of dedicated fields
Resource	Activity executed by unauthorized resource	An activity is executed by a resource other than those authorized
	Activities executed by multiple resources	Activities within the same trace are executed by multiple resources
	Activities executed by a single resource	Activities within the same trace are all executed by the same resource
	Frequent occurrence of activity for a resource	An activity occurs more frequently for one resource compared to other resources
	Frequent occurrence of value for a resource	A data value occurs more frequently for one resource compared to other resources
Time	Occurrence of activity outside of time period	An activity occurs outside of the usual time period
	Delay between start of trace and activity is out of bounds	There is a deviation in the delay between the start of the trace and the time of an activity
	Time between activities out of bounds	There is a deviation in the time between activities
	Duration of activity out of bounds	There is deviation in the duration of an activity
	Duration of trace out of bounds	There is a deviation in the duration of a trace
	Delay between event and logging is out of bounds	There is a deviation in the delay between time of event and time of logging

Acknowledgments

This publication is part of the WorkAround Mining (WAM!) project (with project number 18490) which is (partly) financed by the Dutch Research Council (NWO).

References

[1] T. Ejnefjäll, P. J. Ågerfalk, Conceptualizing Workarounds: Meanings and Manifestations in Information Systems Research, Communications of the Association for Information Systems (2019) 340–363.

[2] S. Alter, Theory of Workarounds, Communications of the Association for Information Systems 34 (2014).

[3] I. Beerepoot, I. van de Weerd, Prevent, Redesign, Adopt or Ignore: Improving Healthcare using Knowledge of Workarounds, in: Twenty-Sixth European Conference on Information Systems, 2018.
[4] I. Ignatiadis, J. Nandhakumar, The effect of ERP system workarounds on organizational control: An interpretivist case study, Scandinavian Journal of Information Systems 21 (2009) 3.
[5] V. Ilie, Psychological reactance and user workarounds. a study in the context of electronic medical records implementations, ECIS (2013).
[6] B. Azad, N. King, Enacting computer workaround practices within a medication dispensing system, European Journal of Information Systems 17 (2008) 264–278.
[7] A. L. Tucker, The Impact of Workaround Difficulty on Frontline Employees' Response to Operational Failures: A Laboratory Experiment on Medication Administration, Management Science 62 (2016) 1124–1144.
[8] S. Weinzierl, V. Wolf, T. Pauli, D. Beverungen, M. Matzner, Detecting Workarounds in Business Processes-a Deep Learning method for Analyzing Event Logs, in: Proceedings of the 28th European Conference on Information Systems, 2020.
[9] I. Beerepoot, X. Lu, I. van de Weerd, H. A. Reijers, Seeing the Signs of Workarounds: A Mixed-Methods Approach to the Detection of Nurses' Process Deviations, in: Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences, 2021.
[10] N. Röder, M. Wiesche, M. Schermann, A Situational Perspective on Workarounds in IT-Enabled Business Processes: a Multiple Case Study., in: ECIS, 2014.
[11] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. P. van der Aalst, Balanced multi-perspective checking of process conformance, Computing 98 (2015) 407–437.
[12] A. Lazovik, M. Aiello, M. Papazoglou, Associating assertions with business processes and monitoring their execution, in: Proceedings of the 2nd international conference on Service oriented computing, ACM Press, 2004.
[13] K. Mahbub, G. Spanoudakis, A framework for requirents monitoring of service based systems, in: Proceedings of the 2nd international conference on Service oriented computing, ACM Press, 2004, pp. 84–93.
[14] A. Rozinat, W. M. P. van der Aalst, Conformance checking of processes based on monitoring real behavior, Information Systems 33 (2008) 64–95.
[15] H. Nguyen, M. Dumas, M. La Rosa, F. M. Maggi, S. Suriadi, Business Process Deviance Mining: Review and Evaluation, ArXiv e-prints (2016). arXiv:1608.08252.
[16] T. Lauer, B. Rajagopalan, Examining the relationship between acceptance and resistance in system implementation, AMCIS 2002 Proceedings (2002) 179.
[17] G. Ras, M. van Gerven, P. Haselager, Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges, in: The Springer Series on Challenges in Machine Learning, Springer International Publishing, 2018, pp. 19–36.
[18] N. Outmazgin, P. Soffer, A process mining-based analysis of business process work-arounds, Software & Systems Modeling 15 (2014) 309–323. doi:10.1007/s10270-014-0420-6.
[19] W. van der Waal, I. Beerepoot, I. van de Weerd, H. A. Reijers, The SWORD is Mightier than the Interview: A Framework for Semi-automatic WORkaround Detection, in: International conference on business process management, 2022. To appear.