Business Process Workarounds: What Can and Cannot Be Detected by Process Mining

Nesi Outmazgin and Pnina Soffer

University of Haifa, Carmel Nountain 31905, Haifa, Israel
Nesi@Zefat.ac.il, spnina@is.haifa.ac.il

Abstract. Business process workarounds are specific forms of incompliant behavior, where employees intentionally decide to deviate from the required procedures although they are aware of them. Detecting and understanding the workarounds performed can guide organizations in redesigning and improving their processes and support systems. Existing process mining techniques for compliance checking and diagnosis of incompliant behavior rely on the available information in event logs and emphasize technological capabilities for analyzing this information. It is therefore not certain that all the forms of workaround behavior are addressed. In contrast, the paper builds on a list of generic types of workarounds found in practice, and explores whether and how they can be detected by process mining techniques. Results obtained for four workaround types in five real-life processes are reported. The remaining two types are not reflected in events logs and cannot be detected by process mining.

Keywords: Business process workarounds, Process mining, Compliance checking

1 Introduction

Business processes are automated and managed by organizations in order to streamline and standardize their operations in an effective and efficient manner. However, the standard prescribed procedures are not always followed by employees, and are many times bypassed and worked around.

Addressing such situations is related to the general area of compliance management, which has drawn much attention in recent years [5]. In the general area of compliance management, several types of activities have been identified [11][12][19]. In particular, compliance checking, which checks whether certain constraints are or will be met, and compliance improvement. Compliance checking can be further divided to forward compliance checking, targeting the design and implementation of processes where compliance is enforced, and backward compliance checking, focused on the detection and diagnosis of non-compliant behavior. Compliance improvement modifies the process to improve compliance. This can be done based on diagnostic information resulting from backward compliance checking, and with the use of forward compliance checking techniques.

This paper focusses on backward compliance checking. Yet, as opposed to the general area of compliance management, which refers both to internal policies and external regulations, the focus of this paper is on situations where employees are aware of the required internal procedures and intentionally decide to act differently. We term these situations business process workarounds. As an example, consider a situation where a customer is urgently requesting some goods and a truck is about to embark in his direction. An employee might decide to immediately load the goods on the truck, while the "paperwork" of registering the order and the delivery will be done afterwards in retrospect.

Workaround are generally considered as a negative phenomenon, assuming the standard process has been designed and optimized to achieve desired business performance. However, since these are intentional actions of employees, we assume they are performed for certain reasons. According to [16], workarounds can be motivated when the defined business processes are rigid and not designed to accommodate situations that might arise, requiring an appropriate response. Additionally, workarounds might be performed when the process design or its support system do not satisfy all the stakeholder needs and expectations. Additional cases might be when employees decide to act upon their own personal goals rather than to follow the defined procedures.

Detecting workarounds and investigating the reasons that drive them can serve organizations striving to compliance improvement and to the design of better processes where workarounds will be reduced. Corrective actions can include process redesign, focused improvement of the business process support system, focused training of the employees, or disciplinary actions.

Various compliance checking techniques have been proposed in recent years as part of the process mining stream of research [1]. These techniques utilize event logs for detecting incompliance to specific constraints, procedures, and process models [3][6][21]. As discussed above, workarounds, as specific forms of incompliance, can be detected using these techniques.

However, the starting point of these techniques is the event log and the technology capabilities. It is therefore not certain that all the forms of workaround behavior are addressed. In contrast, this paper takes a list of six generic workaround types, which were found to exist in business processes [16] as a starting point. Our aim is to explore whether and how workarounds of each of these types can systematically be revealed based on an event log. Building on generic workaround types captures the intentional aspect of workarounds and enables distinguishing them from other types of incompliance. Moreover, it enables us to look for specific patterns that may exist in the log, and to understand what types of workarounds cannot be detected based on the log, if any.

Note that our goal is not to develop new mining techniques. Rather, we wish to explore the capabilities of current technology, commercially available to organizations facing the given workaround types. To this end, we have used Fluxicon Discovery platform (http://fluxicon.com/disco) and applied it to logs of five processes taken from three organizations over two years. To generalize the findings, we further discuss capabilities of state-of-the-art technology for addressing these situations.

The remainder of the paper is structured as follows. Section 2 presents the six generic workaround types identified by [16]. Section 3 discusses the patterns that should be

detected in logs with respect to each of the workaround types; Section 4 reports the findings that were obtained for five real-life processes and discusses them. Related work and available state-of-the-art technologies are discussed in Section 5, and conclusions are given in Section 6.

2 Generic Workaround Types

This section presents the six generic workaround types that were identified by [16] in a qualitative study performed in several organizations.

Type A – Bypass of process parts
In these workarounds, parts of the process are bypassed, so activities that should be performed at later steps of the process are performed before their time. The activities that were bypassed can be performed in retrospect, or skipped altogether.

As an example, consider a purchasing process, where a participant places a purchase order, and only afterwards initiates the formal approval process.

According to [16], this workaround type appears to be common in practice, and is associated with many situational factors that may indicate reasons that drive its performance. Some factors are related to the process support system, e.g., poor user friendliness and a lack of integration among systems. Other factors relate to process design, which can be complicated and cumbersome, hard to understand, involving many different roles, or not in line with the actual needs and the way the process is practiced. Poor information flow and a lack of feedback about the process status to the process initiator, combined with delays and long execution times, are major drivers that motivate employees to commit workarounds of this type.

Type B – Selecting an entity instance that fits a preferable path.
This type of workaround relates to situations where a "legitimate" process execution is performed, but the entity instance that is used does not represent the actual one. Rather, it is chosen in order to comply with the transition conditions of the process. As an example, consider a purchase approval process, where transition conditions require additional approvals if the price is over a certain threshold. Employees who know the rules might split purchase requests, whose price exceeds the threshold, and place several requests, each at a relatively small price, to avoid long approval trails.

Usually, the process participants who perform this type of workaround are experienced and knowledgeable, thus they are familiar with the "rules of the game". Consequently, the workarounds are performed systematically and sophisticatedly. These workarounds are mainly associated with complicated and inflexible transition conditions defined in the process.

Type C – Post factum information changes
This type refers to situations where process participants modify data values after these have been used for decision making. There are two variants of this workaround type.

First, the data modifications reflect values which were known a-priori and falsely entered with the intention of manipulating the decision making.

For example, in a purchase requisition approval process participants give false information (amounts, purchase items, suppliers, quantities, etc.) which allows the process to move "smoothly" and quickly, and only once the approval steps are completed do they change the information to reflect the real needs. Entering the correct information at the initial stage would have required a different path of approvals and control. Similarly to workarounds of the previous workaround type (B), these workarounds are performed sophisticatedly by experienced employees, who exploit loosely defined access control policies and poor authorization management.

The second (less severe) variant of this workaround type is when the modifications reflect new information or error correction, but no re-iteration of the previous decision is made. This can stem from low awareness of the implications of deviating from the required procedures, as well as poor control policies.

Type D – Incompliance to role definition

In this type of workaround, participants perform operations which are not under their responsibility. As an example, consider again a purchasing process, where the initiating participant opens a purchase requisition. When the requisition is approved, it should be handled by a purchasing clerk, who obtains price quotations and selects a winning supplier. A workaround would be when the initiating participant makes inquiries and selects a supplier, and only then transfers the requisition to the purchasing department with the results ready for continued handling.

According to [16], these workarounds typically occur when responsibility assignment does not match (or is not conceived as matching) the knowledge required for certain tasks (e.g., a purchase clerk might not have sufficient technical knowledge to evaluate the available product configurations). Additionally, it might stem from a lack of clear responsibility definitions at different parts of the process. One possible consequence is a poor level of control (incompliance to the "four eyes rule").

Type E – Fictitious entity instances

Workarounds of this type are usually performed by employees to compensate for missing or incomplete process definition and support. When certain process steps or variants exist bur are not managed and supported within the scope of the process, to gain the possibility of monitoring and documentation, fictitious entity instances are created. These instances are marked (e.g., ItemID 99999) and serve the employees for keeping trace of the unsupported parts of the process.

As an example, in a student intake process, it is impossible to perform an acceptance interview with a candidate before he registers (and has a record). However, the candidate might not wish to register before an interview takes place. To overcome this, the secretary creates a fictitious registration in order to continue the process and invite the candidate for an interview. She immediately assigns the candidate to a fictitious room (to mark that the candidate is awaiting an interview).

Although the intention that drives workarounds of this type is to improve the performance of the process, overcoming problems and increasing the level of control, it is still an intentional (and systematic) deviation from the defined procedures.

Type F – Separation of the actual process from the reported one
In this workaround type, at a certain stage the process participants continue the process manually, possibly until the process is completed. At a separate point in time, the actions that were performed (or should have been performed) are reported in an orderly manner. This is done in a post-hoc manner, only for the purpose of documentation and reporting.

An example of this kind of workaround can, again, be found in a purchasing process. Assume a purchase requisition is waiting for a manager's approval. This might take some time, although the chance that the requisition will not be approved is extremely low to non-existent. Facing this, process participants might not wait for the desired approval and rather move forward with the actual process. Once the approval is obtained, the actions that have been performed (e.g., ordering from the supplier) can be recorded in a post-hoc manner.

Situations where such workarounds are performed are characterized by a high number of administrative steps that do not make real contribution or affect the achievement of the process goal, especially if these steps are likely to cause delays and entail long waiting times. It also appears that workarounds of this type are common when the process moves back and forth between organizational units.

3 Detecting Workarounds in an Event Log

This section examines whether the workaround types discussed above can be detected in an event log, and how. We discuss each workaround type and when possible, specify conditions that should indicate its occurrence in an event log.

Type A – Bypassing process parts:
This type is characterized by skipping and bypassing certain process parts. Process instances where such workarounds take place are, hence, incompliant with the prescribed process model, and can be identified using compliance checking techniques. Yet, not every incompliant behavior can be classified as workaround of this type. Specifically, we are looking for activities that are performed while their immediate predecessor (or predecessors) required by the process model have not been performed. The immediate predecessors of an activity can be another activity (if it is in a sequence), several alternative activities (in case the activity follows an OR merge), or several activities that should all be performed (in case the activity follows a synchronization point). We denote the collection of these as PR(a) – the set of immediate predecessors of activity a.

Consider a trace where activity a appears in the *ith* position. If for all $r \in \text{PR}(a)$, r is not included in positions 1..i-1 of the trace, then the trace includes a type A workaround (bypassing process parts). This is checked for all the activities in the trace.

Note that this is a general condition, and it might be too coarse-grained to capture all bypass cases. However, it can be refined and tailored for specific situations. Specifically, it might be required to check the existence of the immediate predecessors of an activity only in part of the trace, after a certain point. For example,

if the process includes loops, the immediate predecessors should be found in the trace between consecutive occurrences of the activity.

Fig. 1 provides an example of a mined model of a purchase requisition approval process, where bypasses are marked (arrows 1, 2, and 3). As an example, according to the required procedures, the immediate predecessors of *Closed* are either *Authorized* or *Declined*. The mined model indicates instances where neither was included in the trace preceding *Closed* (e.g., *Draft ->Closed*, or *Draft ->Auth Process ->Closed*). These are classified as workarounds of Type A.

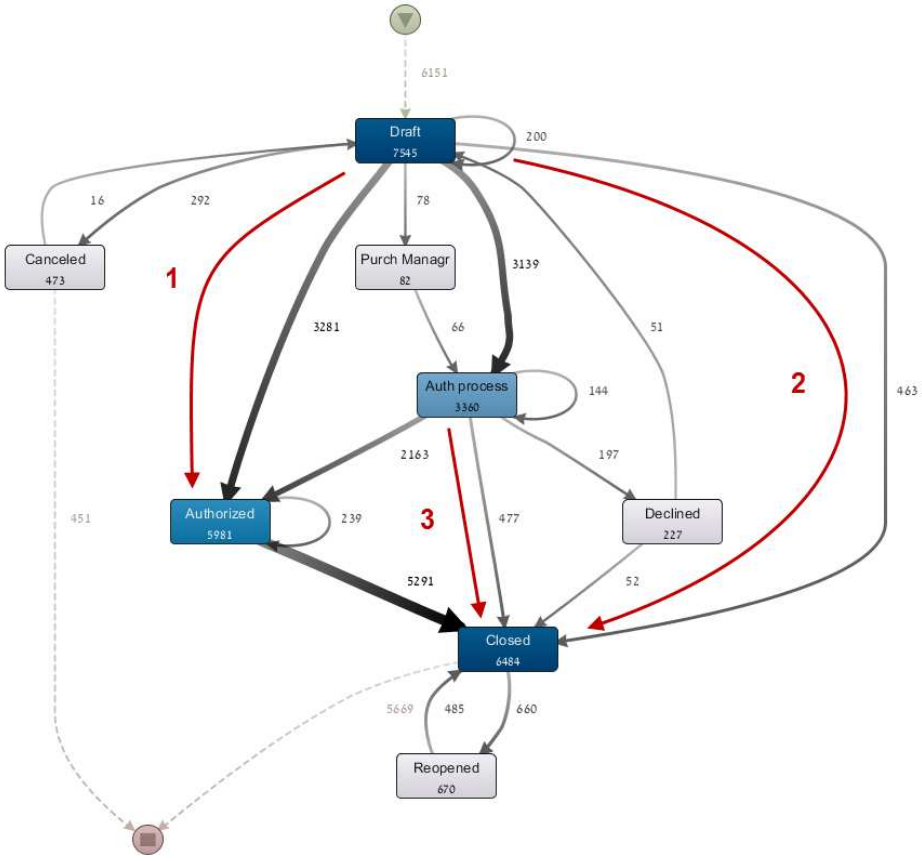

Fig. 1. Bypassing process steps in a purchase requisition approval process (Case Study 3)

Type B – Selecting an entity instance that fits a preferable path:
Process instances where this type of workaround is committed are legitimate instances in terms of control flow. In fact, they might seem legitimate in every process aspect. Yet, they are not accurate reflections of the real life process. Hence, mining event logs cannot detect workarounds of this type. When specific selection types are known to

exist through domain knowledge (e.g., splitting purchase requests), it might be possible to formulate identifiable patterns that would help quantifying these specific behaviors, but these would not be applicable for discovering other cases of this type. It might be possible that data mining techniques aimed at fraud detection (e.g., [17]) can be used for this purpose. However, this is beyond the scope of this paper.

Type C – Post factum information changes:
Workarounds of this type take place at certain stages of the process. Specifically, update data operations are performed after the data has been used by decision making steps (e.g., approval). However, not every modification in the value of a data item that takes place after the data has been used in illegitimate (e.g., errors can be identified and corrected). For a data update to be considered workaround of this type, three conditions should hold:

(1) An update is performed to a data item that has been used previously in the process.
(2) The previous use was for decision making.
(3) After the data update, the process instance does not iterate back to the decision making step (for revisiting the decision based on the updated value).

Clearly, these conditions cannot be directly checked in an event log without additional domain knowledge that would indicate which data is used for decision making at which process steps. Without such indication, skipping reiteration after the update of the data would appear like bypassing process steps (workaround type A).
Using domain knowledge, we can identify data update activities and decision activities relying on the relevant data item.

Consider a data update activity *u*, and let *d* be an activity where this data is used as a basis for decision making. Assume u appears in a given trace in the *ith* position, while *d* can appear in position *j*, $j<i$. If *d* is not included in the trace in position *k*, $k>i$, then this trace includes a workaround of type C.

Note that more than one decision activity might be needed according to the process definition. It should be possible to similarly check the existence of several activities in the remaining part of the trace (at least one or all together).
Area 4 in Fig. 1 provides an example of post factum updates, where purchase requisitions that are already closed are reopened for updating their data (update activity) and then closed again. One related decision activity that should follow reopening is *Authorized*. In the mined process, 485 of the 660 instances that were reopened were then closed (while the remaining ones, which reiterated to approval steps, have been filtered out in the analysis).

Type D – incompliance to role definition: these workarounds are characterized by situations where participants perform activities outside the realm of their responsibility. Apparently, it is easy to detect such workarounds by comparing the user of every activity with the list of users who are permitted to perform it. However, these workaround can only take place if the permissions defined in the system are not tight enough, so unauthorized users can perform the activities. Hence, for accurately detecting these workarounds, the permission assignment should be prepared by the

process owner independent of the existing system permissions. Based on such list, identifying activities that are performed by unauthorized users is straightforward.

Denoting the set of users who are authorized to perform activity a by AT(a), and consider a trace where a is performed by user u. If $u \notin$ AT(a) then the trace includes a workaround of type D.

As an illustration, Table 1 presents the authorized and actual users of activities in a process taken from one of the organizations that were studied. As can be seen, some activities are performed by unauthorized users. In particular, financial approval (3022 out of 3326 times performed by the user P9) and final approval (3065 out of 3303 times performed by P11) are performed by several other users who are not authorized to perform them.

However, it might be that a temporary permission has been granted to, e.g., P8, to perform these activities when the employee responsible for them was away. If that is the case, then along the time, the instances where P8 performed these activities should appear in one or several relatively short periods. This was not found in our case, where the instances involving P8 in these activities were scattered along the two years whose logs were analyzed.

Table 1. Authorized vs. actual users of activities

Activity	**Participant**												**Total**
	Auth	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	
Create PR	all	454	1185	0	223	1	0	175	343	44	1263	0	3688
Manager approval	P1 P10	376	0	0	0	1	0	0	0	0	1121	0	1498
Financial approval	P9	0	38	170	35	16	0	0	44	3022	1	0	3326
Director approval	P11	0	0	0	0	0	0	0	0	0	0	190	190
Buyer approval	all	0	1119	1308	160	26	0	169	311	3	0	0	3096
CEO approval	P5	0	0	0	0	3307	0	0	0	0	0	2	3309
Final approval	P11	1	13	0	2	0	96	9	102	0	15	3065	3303
Cancel PR	all	11	30	9	1	3	0	20	5	3	8	0	90
Close PR	all	356	1109	1	184	0	0	163	341	3	1125	0	3282
Reopen PR	all	0	0	0	0	0	0	0	10	0	1	0	11
Total		1198	3494	1488	605	3354	96	536	1156	3075	3534	3257	21793

Type E – Fictitious entities: in workarounds of this type, a fabricated entity instance is created, to allow the users manage and document process parts that are not included in the formal process (and hence cannot be properly monitored and documented). The resulting process instances appear like legitimate process instances (although they would typically not cover the entire process, but only specific parts).

Following this, mining the control flow of the process would not provide any indication of these workarounds. However, employees who perform workarounds of this type typically mark the fictitious entities by specific codes, so they can distinguish them from real ones. For example, in the student intake process described above, fabricated students were always assigned to Room 1000 (which was fabricated too). If this "marking" information is provided by a domain expert, the relevant process instances can be identified, but this would only serve for quantification of a known phenomenon, not for discovery of unknown ones.

Type F – Separation of the actual process from the reported one: these workarounds entail manual performance of process parts (which cannot be reflected in the log), and reporting the actions to the system just for the record, at some unrelated time. While we cannot tell what actually took place in the (manual) process, the post-hoc recording would usually reflect a "normal" and legitimate process execution, compliant with the required procedures.

Still, we suggest that at least some of these workarounds can be tracked by situations of substantial delays in the process, immediately followed by a bundle of transitions appearing one after the other in an unreasonably short time (as compared to the "normal" process transition times, e.g., three activities performed within a few minutes). For example, consider the instance of a purchase requisition approval process depicted by the log in Table 2. The activity of *Director approval* takes an extremely long time (compared to the activities that precede it), and is followed by two activities whose duration is less than one minute. It is reasonable to believe that the process has in fact progressed before the *Director approval* has been formally given, and that *Approve PR* and Close PR are just reported in a post hoc manner.

Table 2. An example log part demonstrating workaround type F

Activity	**Date**	**Start Time**	**Duration**
Create PR	11.10.2011	12:27:00	9 mins
Buyer approval	11.10.2011	12:36:00	2 hours, 52 mins
Financial approval	11.10.2011	15:28:00	6 hours, 11 mins
CEO approval	11.10.2011	21:39:00	10 hours, 10 mins
Director approval	12.10.2011	07:49:00	15 days, 46 mins
Approve PR	27.10.2011	08:35:00	< 1 min
Close PR	27.10.2011	08:35:00	< 1 min

It can hence be concluded that instances including workarounds of this type might seem legitimate in terms of their control flow, but can be detected based on activity

durations. For each activity *a*, we need to define an upper duration threshold UDT(*a*) and a lower duration threshold LDT(*a*).

For a given trace, if two consecutive activities *a* and *b* are found, such that their durations satisfy d(*a*) ≥ UDT(*a*), and d(*b*) ≤ LDT(*b*), then the trace includes a workaround of type F.

The duration thresholds can be defined based on the log, e.g., by setting a range such that the durations of a defined ratio of the activity instances in the log are above (or below) that range. Note that the upper threshold might even be slightly above the average duration, but the lower threshold needs to be such that the activity cannot possibly be executed within this time. Often, there would be several activities, whose durations are below the lower threshold, performed one after the other. These would be all the activities that have been performed off-line and reported in retrospect.

4 Application to Real Logs

The previous section provided means for identifying four of the six workaround types in event logs. This section reports the results obtained for logs of five processes taken from three organizations over two years. We aimed at addressing processes whose roles are similar in different organizations, as detailed in Table 3.

Table 3. Processes whose logs were analyzed

Process	Title	Organization description
1	Purchase requisition approval	Academic organization, 500 employees
2	Purchase requisition approval	Manufacturer of control and monitoring systems, 300 employees
3	Purchase ordering	
4	Purchase requisition approval	Marketing organization, importing and selling medical equipment, 300 employees
5	Purchase ordering	

As discussed in the introduction, we have decided to use Fluxicon Discovery as a commercial process mining platform, currently available to organizations. The conditions discussed in Section 3 were operationalized using the necessary domain knowledge which was obtained from the process owners. The detailed conditions were then implemented as separate filters over the event logs. Table 4 provides the findings that were obtained. Note that each workaround type was addressed separately, so summarizing all types together would not make sense, since there are instances where more than one workaround type was detected. Moreover, some workarounds can be classified to more than one type. For example, when workarounds of type F (actual process vs. reported one) are performed, often the same person reports several operations, including ones outside his/her role (thus they can also be classified as workarounds of type D).

As can be seen in Table 4, organizations as well as processes within the same organization differ from one another in the frequency of workarounds and in their types. In general, workarounds of type A (bypassing) are the most frequent ones. Difference among organizations is especially evident with respect to organization 1,

whose number of workarounds is extremely low in the purchase requisition approval process. In contrast, in the other two organizations the purchase requisition approval process has a much higher workaround rate than the purchase ordering process. In organizations 2 and 3 the requisition approval process entails a high number of workarounds, especially of type A (bypassing). In organization 2, type D (incompliance to role definition) is also frequent, and in organization 3, types F (actual vs. reported process) and C (post factum information change) are often taken.

Table 4. Workaround percentage by type

Organization	**Process**	**Number of instances**	**% instances with workarounds by type**			
			A	**C**	**D**	**F**
1	PR approval	3688	5.1%	1.3%	2.7%	5.9%
2	PR approval	6920	53.2%	8.8%	22.3%	12.0%
	Purchase ordering	4211	6.8%	7.2%	24.4%	12.6%
3	PR approval	21289	75.3%	25.0%	3.5%	68.1%
	Purchase ordering	5217	11.9%	4.8%	9.0%	4.1%
Average in all processes			30.5%	9.4%	12.4%	20.5%

We note that considering our notion of workarounds, these findings might include both false positives, cases that are falsely indicated as workarounds, and false negatives, actual workarounds that are not detected. Specifically, we define workarounds not just as incompliant behavior, but as one that involves intentional defiance of known procedures. Clearly, we have no means for assessing user intention from event logs. To this end, we rely on the list of workaround types, which was obtained through interviews where users indicated what they perceive as workarounds. It might be that the resulting patterns also include incompliant behavior performed for different reasons.

For example, the cases identified as workarounds of type C (post-factum information change), might include error corrections (where data should be modified to correct the error). According to the regulations, re-iterations to the decision steps (e.g., approval) were required. It might be that this was done informally by emails or phone calls, but the system has no track of these. Hence, officially these cases are considered as workarounds. Similarly, identified cases of type D (incompliance to role definition) might include cases were a temporary permission was granted by the authorized user. We tried to detect these cases by examining the distribution of these occurrences over time. However, one-time permissions cannot be detected this way.

False negatives would relate mainly to types A (bypassing) and F (actual process vs. reported one). Bypasses (type A) can be performed manually (e.g., ordering goods by phone) and not reported, while the process as reflected in the log appears to progress according to the required procedures. Considering separation of the actual process from the reported one (type F), our detection method is based on the assumption that this can be reflected in the log as exceptional durations of activities

(exceptionally long duration of one activity followed by one or more exceptionally short durations). This assumption does not necessarily apply in all cases. Specifically, the post-hoc reporting might be performed at different points in time for different activities, which would not appear as exceptional activity durations.

Still, even with these limitations, we believe that quantification like the one in Table 4 is valuable for organizations. In particular, it can serve as a starting point for investigating the workarounds that are performed and lead to corrective actions that should address the reasons that drive these workarounds. The result of such actions should be improved processes with improved compliance.

Finally, we note again that two types of workarounds were not possible to detect from the logs, yet they are likely to exist. Being aware of this possibility, organizations can apply targeted means for identifying and addressing them. Fictitious entities (type E), for example, usually involve practices which are well known among the relevant users, sometimes even anchored in departmental documents and procedures. Typically, they are marked by specific IDs that would enable the users to track them. It should hence be rather easy to specifically elicit them from the employees and make appropriate modifications to the process. Intentionally selected entity instances (type B) would be more difficult to expose, especially since these are performed by sophisticated employees with the intention to avoid the required process paths. As discussed, data mining techniques might be of assistance.

5 Related Work

While much attention has been given to compliance management in general and compliance checking in particular, the specific phenomenon of intentional workarounds has not been extensively investigated. Nevertheless, the conditions defined here in correspondence with workaround types can be verified by some of the existing compliance checking approaches. This section reviews the relevant literature, indicating the workaround types that can be detected by each approach.

Several approaches have been suggested for backward compliance checking. These include replaying-based techniques (e.g., [21][4][6][7][8]), where a process is replayed on the log against the required process model, and rule checking techniques, where rules can be defined using Linear Temporal Logic (LTL) [3][13] or Petri net representation [9][10][14][15][18]. Replaying-based techniques address incompliant behavior in general, as opposed to the specific set of behaviors we address in this paper. Behavior types that would be detected by these techniques include some of the workaround-related patterns, as well as additional ones, such as activity repetition, or performance of additional or different activities as compared to the process definition. In contrast, rule-based conformance checking can relate to specifically defined rules (including those related to workarounds). Hence, we focus on this group of approaches.

[18] define 15 categories of control-flow compliance rules. Four of these categories are relevant in our context of workaround detection. *Existence rules* limit the occurrence or absence of a given event within a scope – these can be used for identifying workarounds of type A (bypassing) and of type C (post-factum

information change). *Precedence rules* require or limit the occurrence of a given event in precedence to another event – these can be useful for detecting workarounds of type A, since a violation of such rule implies that activities have been bypassed. *Response rules*, which require or limit the occurrence of a given event in response to another event – can be used for detecting workarounds of type C, where a post-factum information change is considered as workaround only if it is not followed by reiteration of decision steps. *Between rules* require or limit the occurrence of a given event between two other given events – can be used for detecting bypassing (type A) in a process which includes loops.

These compliance rules can be checked by LTL-based approaches [3][13], which are easily capable of specifying these kinds of constraints. Petri net-based methods specify a rule as a Petri net segment, and then find a best alignment with the log [2][18][19]. While LTL-based rules address only the control flow of the process, and are thus relevant for detection of the two aforementioned workaround types (A and C), the alignment seeking Petri net based approaches can handle other aspects as well.

[18] address compliance to data and organizational aspects, which enables detecting workarounds of type D (incompliance to role definition). The data-related techniques are extended in [19] to address temporal constraints, which are capable of capturing the exceptional activity durations that characterize workarounds of type F (actual vs. reported process). It can hence be concluded that the alignment-based methods provide powerful means that enable specifying appropriate rules and detecting the four workaround types that are reflected in event logs.

6 Conclusion

Workarounds are often performed in business processes. Compliance management literature has not addressed them as a distinct phenomenon so far, but rather as part of incompliant behavior in general. We believe that intentional defiance of known procedures should receive special attention, since revealing this behavior and the reasons that motivate it can expose many underlying problems that need to be solved.

A main contribution of this paper is in approaching this issue from a practice perspective. As opposed to existing works in the area of compliance checking, which focus on the capabilities of technology to be utilized, this paper departs from behavior types that exist in practice and are perceived by employees as intentional workarounds. It uses six generic behavior types identified in organizations, and seeks for technological solutions that can serve for detecting these behaviors. It does so by analyzing and characterizing the log patterns that can be associated with the considered workaround types. We have specified conditions that enable detecting four workaround types in event logs and demonstrated their ability to quantify the occurrence of each type in logs of five real-life processes.

An important finding is the indication of two workaround types that leave no recognizable trace in the log and hence cannot be generically identified by process mining techniques. Still, additional domain knowledge can be used for defining specific patterns that might be identified in logs. This highlights the limitations of

generic process mining techniques and can guide organizations in further directions that need to be taken to completely address the workaround phenomenon.

Developing an understanding of the workarounds that take place and particularly of the reasons that drive them would be valuable in improvement efforts. Corrective actions can include redesigning the processes, improving the data flow, the permission and control mechanisms, role definitions, and also training and disciplinary actions. This is expected to lead to improved performance as well as compliance.

Future research will aim at investigating the reasons for workarounds, and establish relationships between process properties, such as bottlenecks and number of participants, and the frequency of workarounds.

References

[1] Aalst, W.M.P.v.d.: Process Mining – Discovery, Conformance and Enhancement of Business Processes. Springer (2011)

[2] Aalst, W.M.P.v.d., Adriansyah, A., Dongen, B.F.v.: Replaying history on process models for conformance checking and performance analysis. WIREs Data Mining Knowledge Discovery 2, 182–192 (2012)

[3] Aalst, W.M.P.v.d., Beer, H.T.d., Dongen, B.F.v.: Process mining and verification of properties: An approach based on temporal logic. In: OTM Conferences 2005. LNCS, vol. 3760, pp. 130–147. Springer (2005)

[4] Aalst, W.M.P.v.d. Business Alignment: Using Process Mining as a Tool for Delta Analysis. In J. Grundspenkis and M. Kirikova, editors, Proceedings of BPMDS'04, volume 2 of Caise'04 Workshops, p. 138–145 (2004)

[5] Abdullah, N.S., Sadiq, S.W., Indulska, M.: Information systems research: Aligning to industry challenges in management of regulatory compliance. In: PACIS 2010. p. 36. AISeL (2010)

[6] Adriansyah, A., Dongen, B.F.v., Aalst, W.M.P.v.d.: Towards Robust Conformance Checking. In: BPI 2010. LNBIP, vol. 66, pp. 122–133. Springer (2011)

[7] Adriansyah, A., Sidorova, N., Dongen, B.F.v.: Cost-based Fitness in Conformance Checking. In: ACSD 2011. pp. 57–66. IEEE (2011)

[8] Adriansyah, A., Dongen, B.F.v., Aalst, W.M.P.v.d.: Conformance checking using cost-based fitness analysis. In: EDOC 2011. pp. 55–64. IEEE (2011)

[9] Calders, T., Guenther, C., Pechenizkiy, M., Rozinat, A.: Using Minimum Description Length for Process Mining. In: SAC 2009. pp. 1451–1455. ACM Press (2009)

[10] Fahland, D., de Leoni, M., van Dongen, B.F., van der Aalst, W.M.P.: Conformance Checking of Interacting Processes with Overlapping Instances. In: BPM. LNCS, vol. 6896, pp. 345–361 (2011)

[11] Kharbili, M.E., Medeiros, A.K.A.d., Stein, S., Aalst, W.M.P.v.d.: Business process compliance checking: Current state and future challenges. In: MobIS 2008. LNI, vol. 141, pp. 107–113. GI (2008)

[12] Kharbili, M.: Business process regulatory compliance management solution frameworks: A comparative evaluation. In: APCCM 2012. CRPIT, vol. 130, pp. 23–32. ACS (2012)
[13] Montali, M., Pesic, M., Aalst, W.M.P.v.d., Chesani, F., Mello, P., Storari, S.: Declarative Specification and Verification of Service Choreographies. ACM Transactions on the Web 4(1), 1–62 (2010)
[14] Munoz-Gama, J., Carmona, J.: A Fresh Look at Precision in Process Conformance. In: BPM 2010. LNCS, vol. 6336, pp. 211–226. Springer (2010)
[15] Munoz-Gama, J., Carmona, J.: Enhancing Precision in Process Conformance: Stability, Con-fidence and Severity. In: CIDM 2011. IEEE (2011)
[16] Outmazgin, N.: Exploring Workaround Situations in Business Processes, In: BPM 2012 Workshop proceedings, LNBIP 132, pp. 426-437, Springer (2013)
[17] Phua, C., Lee, V., Smith-Miles, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research, Artificial Intelligence Review (2005)
[18] Ramezani, E., Fahland, D., Aalst, W.M.P.v.d.: Where Did I Misbehave? Diagnostic information in compliance checking. In: BPM 2012. LNCS 7481, pp. 262-278. Springer (2012)
[19] Ramezani, E., Fahland, D., van Dongen, B., Aalst, W.M.P.v.d.: Diagnostic Information for Compliance Checking of Temporal Compliance Requirements, Proceedings of CAiSE 2013, Springer (2013)
[20] Ramezani, E., Fahland, D., Werf, J.M.E.M.v.d., Mattheis, P.: Separating compliance management and business process management. In: BPM Workshops 2011. LNBIP, vol. 100, pp. 459–464. Springer (2012)
[21] Rozinat, A., Aalst, W.M.P.v.d.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)