Categorizing Identified Deviations for Auditing

Marzie Hosseinpour and Mieke Jans

Hasselt University, Agoralaan Gebouw D, 3590 Diepenbeek, Belgium

Abstract. Currently, financial statements auditors perform the tests of controls based on sampling. However, when using a sampling approach, information is lost. To counter this drawback, data analytics has been applied as a method for auditors to provide assurance while using all data. Specifically for testing controls, the potential of process mining has been explained in literature. Indeed, conformance checking can be used to compare real process executions with a normative model. However, the outcome of current conformance checking techniques is too vast for an auditor to inspect further. The identified deviations are at an atomic level (skipped and inserted tasks) and there is no feasible approach to gain a quick overview of the deviations. In this paper, we propose an approach to categorize deviations, which enables auditors to quickly gain an overview of different types of existing deviations along with their frequencies. Categorizing deviating process instances can also give an insight for assessing the risk at case level.

Keywords: Deviation Identification, Financial Statemenets Auditing, Process Mining, Conformance Checking, Risk Assessment

1 Introduction

The objective of financial audit practice, for auditors is to express their opinion on correctness and fairness of organizations' financial statements. To achieve this goal, auditors test business process's controls effectiveness and perform substantive tests of balances and accounts which impact the financial reporting. Test of process controls should be done before investigation of balance sheets and accounts. The rationale is that if the control settings are rigid, there is reasonable assurance, for the auditor, to rely on the organization's internal controls. If the results of tests of controls are not satisfactory, more substantive evidence should be gathered. That means more time and effort should be dedicated while testing the balance sheets and statements. To test the controls, as required by Audit Standard No.5 [2] and ISA 315 [1] a general approach is to first collect information on the process by interviewing process experts and by studying the normative model (or creating one, in case it does not exist). After a general understanding of the normative model, auditors test the effectiveness of the associated control setting. Currently, this is tested by taking a sample of process executions. The sample is compared with the business model manually to check the conformity of the selected cases. If there are no deviating cases among this selection, the control settings are assumed to be reliable. Otherwise, when deviating cases are discovered they should be reported. However, before reporting, the deviations should be studied by auditors and further discussed with process experts. The reason is that some of the deviating cases can be 'cleared' due to some implicit or explicit exceptional rules and be labeled as normal cases. The deviations which have a higher risk level should be prioritized for follow-up. Investigating deviations in such a way is a roughly feasible task owing to the currently used sampling approach. However, some information is lost and results may be inaccurate using sampling.

The advent of process mining techniques [5], in the last decade, can be promising for auditing, not in the least because of its full-population testing ([8,9]). By applying a conformance checking technique (such as [3, 4, 6, 10, 11]), the entire set of real process executions (aka event log) can be compared with the process model to distinguish deviating cases from normal cases. Although these techniques are able to locate the root cause of each deviation, for auditing purpose there are some shortcomings. First of all, the output of detected deviations is too immense for a full follow-up. There are too many variants of deviations. This is because the normative model does not cover all possible exceptional or flexible behavior of processes. The second problem is that almost all of the conformance checking techniques discover deviations in a very atomic level, which is not pragmatic to work with. For example, consider a procurement process with the model <*Create PO, Sign, Release, IR, GR, Pay*> where *IR (Invoice Receipt)* and *GR (Goods Receipt)* are concurrent activities and can change order. Take <*Create PO, IR, Release, Pay*> as an executed trace. If we check the conformity of this trace manually, we would intuitively notice that the activities *Sign* and *GR* are missing in this execution and *Release and IR* have changed order (since *IR* is expected to occur after *Release*, based on the model). By giving the model and the trace to the mostly employed conformance checking tool [6], the output will be as follow: *Skipped(Sign), Skipped(Release), Inserted(Release), Skipped(GR)*. In a real event log, with thousands different cases, this leaves auditors with hundreds combination of low-level deviations, isolated from the variants and the context where they took place.

The question is how the idea of conformance checking (comparing a log with a model) can be made actionable for auditors. The approach we consider in this paper is to create a different layer of deviations that would allow us to categorize those deviations in sets that are meaningful and manageable for an auditor. Concretely, we would like to develop (or alter) a conformance checking algorithm that identifies deviations in such a way that i) they will be meaningful for auditors, ii) gives them an insight of different types of existing deviations and their frequencies in the executed behavior, and iii) helps auditors to prioritize deviations based on their risk level.

In the remainder of this paper, we propose an efficient interpretation of deviations in section 2, which can answer the above question. In section 3, we explain how we will implement the idea and we conclude in section 4.

2 Proposed Approach

To address the problem of too fine-grained deviation types (skipped and inserted) mentioned in the previous section, we define a set of six deviation types that we presume would be meaningful to an auditor, even without the context of the complete trace. These deviation types interpret deviations at a higher level, while keeping the location and root cause of the mismatches in each case. The six types that we propose are: missing a sequence, existence of an extra sequence, loop on a sequence, repetition of a sequence, swapping two sequences, replacing one sequence by another sequence. Note that a sequence can contain one or more activities. These proposed types are explained in table 1 due to lack of space. For each deviation type an example is provided, based on the procurement model presented in section 1.

	Type	Description	Example
1	Missing a sequence	A sequence of events that should have taken place, is not executed	<*Create PO, Release, IR, GR, Pay*> Sign is missing
2	Existence of extra sequence	A sequence of events is executed that it was not designed	<*Create PO, Sign, Change Line, Release, IR, GR, Pay*>*)* Extra event Change Line exists
3	Loop on a sequence	A sequence of events is repeated while only a single occurrence was designed	<*Create PO, Sign, Release, IR, GR, Pay, Pay)*> Loop on Pay
4	Repetition of a sequence	An executed sequence is repeated later in another part of trace	<*Create PO, Sign, Release, IR, GR, Sign, Pay)*> Repetition of Sign after GR
5	Swapping two sequences	Two sequences have changed their order	<*Create PO, IR, Sign, Release, GR, Pay)*> <Sign, Release> and IR are swapped
6	Replacing one sequence by another sequence	A sequence takes place instead of another missing sequence.	<*Create PO, Sign, Pay, IR, GR, Pay*> Release is replaced by Pay

Table 1: Deviation types from control-flow perspective with description and an example from procurement process explained in section 1 for each deviation type

The idea is to develop an algorithm that provides auditors with an overview of deviations according to their categories. Next, it should be feasible to drill down to the sequences that were subject to the deviations. For instance, the deviating trace <*Create PO, IR, Release, Pay*> will be described as follow: Missing an event (with two sub-categories: *Sign is missed* and *GR is missed*) and Swapping events (*Release and IR are swapped*). On a log level, for example, this could give: *"This logs shows 500 times a "Repetition of a sequence", this is stemming from 150 repetition of <Sign, Release>, 150 repetition of IR, and 200 repetition of GR."* which describes existing deviations in the log in categories along with their frequencies.

Therefore, the contribution of this paper is to interpret deviations using these types, which enables auditors to perceive different types of devisions and possibly related risk, as one sees them intuitively (rather than on an atomic level).

3 Methodology and Implementation

Before the development of the desired algorithm, we will perform a field research. The methodology is to interview auditors to test our proposed deviation types in their approach. The field research will be executed to gain insights in how complex deviation types might be interpreted by human experts (as apposed to our assumptions). Consider again, the deviating example in section 1, *<Create PO, IR, Release, Pay>*. The deviation type *'swapped Release and IR'* can be interpreted in a different way like *'Release is postponed after IR'* or even *'IR is advanced before Release'*.

When various deviation types exist in one trace, the combination of them also can be interpreted differently. Avoiding different interpretation of deviations is important because it may lead to assigning wrong risk level to them. Hence, we believe performing the field work research is necessary before implementation of the idea. After the field research, we plan to build the algorithm with the desired requirements. Current known algorithms have already been investigated to what extend they could help in achieving our goal. The shortcoming of cost-based conformance checking tool, proposed by Adriansyah et al. [6] is discussed in section 1. The conformance technique proposed by Garcia et al. [7], to the best of our knowledge, is the only conformance checking tool which provides the deviations in natural language statements. The tool finds deviations in both model and event log. The output is suitable to improve the model or see what types of deviations exist in the event log in general. Nevertheless, their approach is not fully compatible with auditing purpose of testing the controls. The reason is that it does not have the means for finding which cases or traces cause the discovered deviations. Hence, one is not able to find the deviating cases or even the frequency of each deviation type. After the field research, between these two techniques, we will choose the one which is closer to our objective of capturing and categorizing deviations and will adapt it to accomplish our goal.

4 Conclusion and Future Work

This paper introduces the idea of developing a technique for organizing the identified deviations in event logs into certain categories. A set of six different deviation types are proposed to enable auditors to gain an overview of existing deviations.

During our field research, we will also study what type of information auditors use to assess the risk of deviation types. This insight will be used in a follow-up phase to go from deviation to risk classification. Moreover, the correlation between risk level and the complexity of each deviating trace (i.e., the number and the variety of the mismatches in each deviating trace), in a real life setting will be investigated.

References

1. Iaasb: International international federation of accountants, international standard on auditing 315, http://www.ifac.org/system/files/downloads/a017-2010-iaasb-handbook-isa-315.pdf
2. Pcaob: The public company accounting oversight board, auditing standard no. 5, https://pcaobus.org/Standards/Auditing/pages/auditing_standard_5.aspx
3. van der Aalst, W.M.P., de Beer, H.T., van Dongen, B.F.: Process mining and verification of properties: An approach based on temporal logic. In: Proceedings of the 2005 Confederated International Conference on On the Move to Meaningful Internet Systems. pp. 130–147. OTM'05, Springer-Verlag (2005)
4. van der Aalst, W.M.P., de Medeiros, A.K.A.: Process mining and security: Detecting anomalous process executions and checking process conformance. Electron. Notes Theor. Comput. Sci. 121, 3–21 (2005)
5. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Publishing Company, Incorporated, 1st edn. (2011)
6. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Conformance checking using cost-based fitness analysis. In: Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference. pp. 55–64. EDOC '11, IEEE Computer Society, Washington, DC, USA (2011)
7. Garcia-Banuelos, L.and van Beest, N., Dumas, M., La Rosa, M.: Complete and interpretable conformance checking of business processes. BPM center (2015)
8. Jans, M., Alles, M.G., Vasarhelyi, M.A.: A field study on the use of process mining of event logs as an analytical procedure in auditing. The Accounting Review 89(5), 1751–1773 (September 2014)
9. Jans, M., Alles, M.G., Vasarhelyi, M.A.: The case for process mining in auditing: Sources of value added and areas of application. International Journal of Accounting Information Systems 14 14, 1–20 (March 2013)
10. Ramezani, E., Fahland, D., van der Aalst, W.: Where did i misbehave? diagnostic information in compliance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) International Conference on Business Process Management (BPM 2012). Lecture Notes in Computer Science, vol. 7481, pp. 262–278. Springer-Verlag, Berlin (2012)
11. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)