Understanding Process Behaviours in a Large Insurance Company in Australia: A Case Study

S. Suriadi[1], M.T. Wynn[1], C. Ouyang[1], A.H.M. ter Hofstede[1,2], N.J. van Dijk[2]

[1] Queensland University of Technology, Brisbane, Australia
{s.suriadi,m.wynn,c.ouyang,a.terhofstede}@qut.edu.au
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
{n.j.v.dijk}@student.tue.nl

Abstract. Having a reliable understanding about the behaviours, problems, and performance of existing processes is important in enabling a targeted process improvement initiative. Recently, there has been an increase in the application of innovative process mining techniques to facilitate *evidence-based* understanding about organizations' business processes. Nevertheless, the application of these techniques in the domain of finance in Australia is, at best, scarce. This paper details a 6-month case study on the application of process mining in one of the largest insurance companies in Australia. In particular, the challenges encountered, the lessons learned, and the results obtained from this case study are detailed. Through this case study, we not only validated existing 'lessons learned' from other similar case studies, but also added new insights that can be beneficial to other practitioners in applying process mining in their respective fields.

Keywords: process mining, case study, business process management

1 Introduction

Improving business processes has been a top priority for CIOs [3]. Having a reliable understanding about the behaviours, problems, and performance of existing (i.e. 'as-is') processes is important to ensure a targeted improvement initiative [7]. Recently, there has been an increase in the application of *process mining* [15] (a discipline that emerged in the late 1990s) in many domains to enable understanding about existing processes. By relying on data collected in today's information systems, as opposed to subjective perceptions (e.g. interviews), as the starting point for analysis, process mining offers an *evidence-based* approach to derive valuable insights from an organization's processes. Through process mining, one can *discover* the actual 'as-is' process models describing the way processes were executed. Process mining can check the *conformance* of processes against some 'ideal' behaviours (e.g. as documented in standards or guidelines). Furthermore, it can also identify performance issues in processes (such as bottlenecks) and analyze the interaction between resources [20]. Insights gained from process mining analyses can then be used to precisely identify process improvement opportunities.

In recent years, process mining techniques have been applied in a number of organisations, most notably in Europe [15]. Nevertheless, the application of process mining techniques in the domain of finance within Australia is scarce. This paper details a 6-month case study in the application of process mining (and other complementary techniques) in analysing unstructured insurance claims processes in Suncorp, one of the largest insurance companies in Australia. The objective of this case study was to gain insights into the ways that insurance claims were processed at Suncorp by analysing the event log data recorded within their claims management system. In particular, we were interested in process improvement ideas on how to reduce the lengthy claims processing time for a group of seemingly 'simple' claims. The outcomes from this case study could potentially be used to inform the management team in refining their claim processing triage rules such that 'simple' claims could be further streamlined and automated. This was expected to free up more resources to work on 'complex' or high-priority claims with the ultimate aim of improving customers' satisfaction.

The main contribution of this paper is a detailed explanation of the challenges and lessons learned in the use of process mining techniques and methodology to understand and improve the efficiency of the insurance claims processes. The usefulness of process mining in supporting a deeper data mining analysis to explain the reasons for the occurrence of the undesirable lengthy claims processing times is also reported. Through this case study, we not only validate existing 'lessons learned' (drawn from similar case studies [8, 11, 18]), but also add new insights which are expected to benefit other practitioners in applying process mining in their respective fields.

This paper is arranged as follows: Section 2 describes our case study approach. Section 3 details our step-by-step approach to conducting the case study - relevant challenges, lessons learned, and results from each stage of the case study are detailed. Section 4 provides some recommendations on conducting a process mining project based on the lessons learned. Section 5 summarizes a number of related work, while Section 6 provides the conclusion.

2 Approach

A number of approaches have been proposed in the literature to perform process mining in practice [10, 15, 21]. Among these, the most well-known methodology is the L^* *life-cycle model* [15] developed by van der Aalst. In our case study, we decided to follow this approach since it has evolved as a result of many process mining case studies carried out so far. From initial observations of the event logs and interviews with stakeholders regarding the insurance claims processes for investigation at Suncorp, we believe that these processes are messy and unstructured (a.k.a. "spaghetti processes"). Hence, we adopted a variant of the L^* process mining approach, which was also proposed by van der Aalst to cater for analysing "spaghetti processes" [15, Chap. 12]. There are five stages (see Fig. 1).

The *first* stage aims to develop a deep understanding of the business problem. In-depth interviews with the different stakeholders should be conducted to pro-
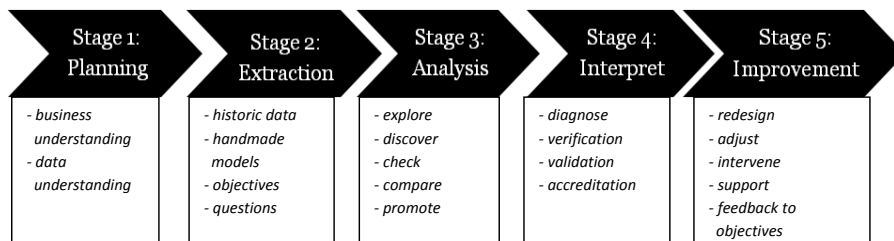
Fig. 1: A simplified L^* process mining methodology (adapted from [15])

vide the necessary business and data understanding [18]. This stage is necessary for all subsequent stages and should not be underestimated.

The *second* stage involves extracting the data from outside sources and transforming (e.g. filtering, correlating, and aggregating) the (event) data for analysis. This is recognized as the most important and challenging task in order to get meaningful insights into process behaviours [19]. Also the questions and objectives (and possibly existing process models) should be identified from interviews with domain experts and analyses of existing systems [15].

The *third* stage focuses on the application of various process mining techniques (e.g. discovery, conformance and enhancement) using the data obtained from the previous stage.

The *fourth* stage aims at deriving insights from the results obtained thus far and to explain the results in the context of the domain. For example, the results may identify certain problems in the existing processes. The diagnoses of these problems can be carried out in a manner similar to root cause analysis (RCA). To this end, we follow the *process log-based RCA* approach [14] which makes use of the well-established *classification* technique from the field of data mining [15, Chap. 3]. The basic idea is to first categorise each case based on its performance indicator to investigate (e.g. short vs. lengthy case processing time) and this indicator is known as a *response* variable. Next, the factors that may contribute to the performance of a case (e.g. insurance brand, resource expertise) are known as *predictor* variables. A classification analysis will then use the information to infer any causal relationship, expressed as *rules*, between the predictor variables and the response variables.

Lastly, the *fifth* stage is to apply the findings so far to the improvement of the existing processes.

3   Case Study: Challenges, Lessons Learned, and Results

The case study was conducted at Suncorp, a leading insurance company in Australia with millions of customers. Suncorp's claims processing system is supported by a mixture of legacy and modern workflow-like systems. To improve customer satisfaction, Suncorp is interested in reducing the overall claim pro-

cessing time by improving their business processes. The results from this case study are expected to assist them in identifying key areas to improve.

This case study is made up of two phases: the first phase was quite 'explorative' while the second phase was more focused due to the clear formulation of *process mining* questions (detailed in Section 3.1). To simplify the presentation, this section reports the overall challenges, lessons learned, and results from both phases. The distinction between the two phases is highlighted when and where necessary. General challenges, lessons, and results are summarized in Section 4, while those specific to each stage of the L^* methodology are detailed in Sections 3.1 to 3.5. Due to practical reasons, deviations from the L^*-methodology occurred and the consequences of these are explained in Section 4.

3.1 Planning

The planning stage in our case study was iterative, it ran alongside other stages, including data extraction, pre-processing, and analysis. It consisted of presentations about process mining to the stakeholders, as well as discussions on how process mining could benefit the stakeholders' organization. Through these engagements, we learned that Suncorp maintained a rich event log capturing their insurance claims processes. This log is suitable for process mining analysis as key attributes (e.g. case ID, activities, and timestamps) are available. A mutually-beneficial engagement model between Suncorp and our university was thus adopted: we obtained rich industry-based data that would be valuable to support our research, and in exchange, Suncorp gained objective and non-biased insights about their business processes from our analysis of their data.

Throughout the case study, a close engagement with the stakeholders (i.e., business analysts) was maintained via (mostly) weekly meetings. These meetings were used to report on the progress of our analysis, to discuss the problems encountered, to gain a better understanding about the data, to request for further data (when needed), and to agree on short-term targets.

The *main* question that Suncorp had which drove the direction of this case study was: *why did the processing of certain 'simple' claims take an unexpectedly long time to complete?*

Challenge. The main challenge during this stage was to define the concept of a 'simple' claim (i.e. to identify the characteristics of a 'simple' claim). Such a definition is crucial as it was necessary to differentiate those cases in the logs that belonged to the 'simple' category from those that did not. Having a clear definition of this concept, and how it was reflected in the data, also facilitated the transformation of the *main* question into specific *process mining* questions which form the starting point for the subsequent data extraction and process mining analyses. This was a challenging process because there were no corresponding business rules behind this concept. Thus, we considered all possible factors in the definition, such as the claim types, the total number of events, and the duration of claims. Through multiple rounds of discussions and the involvement of a process improvement expert, we managed to formulate an acceptable

definition of 'simple' claims. This process took more than five interview sessions and multiple rounds of preliminary analyses over a period of six weeks.

Lessons Learned. Having a team with complementary skills is crucial. Initially, a communication gap existed between the business-oriented stakeholders and the more technically-oriented process mining analysts, e.g. unfamiliarity with domain-specific jargon (e.g. 'incurred amount', 'recovery') and mis-interpretation of stakeholders' goals - a situation that contributed to the challenge described above. In the second phase, we engaged a process improvement expert whose involvement significantly bridged the communication gaps, and enabled a smoother crystallization of the concept of a 'simple' claim and its mapping to the data.

The use of the *question-driven* approach to process mining analysis is very important (as suggested in the L^* methodology [15, Chap. 12]), especially for organizations with limited process mining experience. Our initial approach was *data-driven* [15, Chap. 12]: basic process mining analyses were conducted and the results were then studied for their usability to answer the *main* question. However, since the link between the analyses conducted and the *main* question was not obvious, the findings turned out to be with limited use. After following a clear definition of the concept of 'simple' claims, we switched our approach to a 'question-driven' approach that resulted in more targeted and useful findings.

Result. The agreed definition of a 'simple' claim is a claim whose net payout value is less than x dollars and should be completed no later than y-number of days. However, Suncorp was also aware of some irregularities in the processing of their 'supposedly' simple claims: some of them completed in longer than y-number of days. Based on these insights, we derived three *process mining* questions:

- **Q1:** What is the performance distribution of 'simple' and 'non-simple' (termed 'complex') claims?
- **Q2:** What do the process models look like for 'simple' and 'complex' claims? What are the frequently-taken paths in the processing of these claims?
- **Q3:** What are the key differences in the way claims were processed between those 'strictly' simple claims (completed on-time), and those 'supposedly' simple claims (completed longer than y-number of days)?

Q3 addressed the *main* question by attempting to explain the duration of 'simple' claim processing as a function of process behaviours (detailed in Section 3.4). To enable this, insights gained from Q1 and Q2 were utilised. The details on how we addressed Q1 and Q2 are provided in Section 3.3.

3.2   Data Extraction and Pre-processing

There were two rounds of data extraction and pre-processing: results from the first phase highlighted data quality issues (e.g. omission of important claim attributes), triggering another round of data extraction and pre-processing. There are hundreds of thousands of events in both data sets. The data sets received
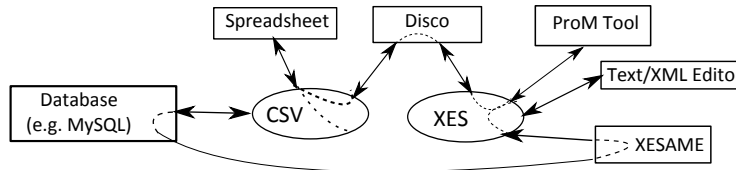
Fig. 2: Various Tools Used in Data Filtering

were for claims that were *finalized* during a certain six-month period (*no running/incomplete cases* in the data sets). The data sets were extracted from a number of Suncorp's insurance claim sub-systems by the relevant IT personnel, and were passed on to us via the business analysts involved. The data sets received were in the comma-separated values (CSV) format.

Once the data was cleaned from potentially-problematic characters (such as ',', '<', '>', and '&'), the data was imported into a database system, and a dedicated process log filtering tool, such as Disco[3], for further data manipulation and filtering. Importing the data into a database system was needed to allow richer data manipulation and analysis. For example, the 'count' and 'GROUP BY' aggregation functions available in SQL queries proved to be very useful in understanding a claim's characteristics (e.g. the number of times the 'follow up' activity was conducted within each claim). On the other hand, the Disco tool was very useful to conduct basic commonly-used data filtering activities, such as filtering data based on case variants and total number of activities.

The filtered data was then converted into standard XES format[4] using either the XESame tool [23] or Disco. As will be explained later, there are certain advantages and disadvantages with the XESame tool and Disco. In this case study, both tools were used.

Challenges. It was difficult to get to a common understanding of the meaning of data and of their most appropriate use for analysis between the stakeholders, the IT personnel, and the process mining analysts due to the complexity of the processes captured in the data (e.g. multiple sub-processes), the high-level of noise (e.g. infrequent process variants), the incomplete recording of events, and the inconsistent usage of certain activities or attributes among the system users (e.g. the inconsistent use of *loss cause* terminology, such as 'natural hazard', 'storm flood', and 'flood', in the processing of claims). Extensive discussions with stakeholders were needed to understand how to use the data for analysis.

Determining the right data filtering criteria was challenging, but necessary to ensure scalable analysis (given the size of the logs) and to enable interesting comparative analysis between data 'slices'. This challenge was related to the first stage (planning) when the definition of a 'simple' claim was not yet crystallized, resulting in ad-hoc filtering of data using somewhat 'random' criteria (e.g. the number of events per case and policy type). This challenge was addressed as soon as the concept of 'simple' claim became clear.

---

[3] http://fluxicon.com/disco/
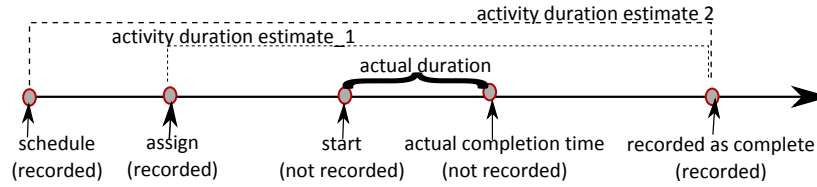
[4] http://www.xes-standard.org/

Fig. 3: The (non-)recording event lifecycle and its impact on duration estimate.

The use of multiple tools also gave rise to compatibility problems: the different requirements for data import among tools (e.g. timestamp format requirements), and the 'quirkiness' in the way data is exported by different tools (e.g. how the empty string is exported) required frequent manual 'massaging' of the data that was time-consuming, tedious, and error-prone.

Lessons Learned. Naive interpretation of data is likely to result in meaningless findings, e.g. a naive interpretation of the activity completion timestamp recorded in the log will treat it as *actual completion time*. However, as we found out, it was actually the time the activity was *recorded as complete* (see Fig. 3). The time gap between these two events can span hours or days and can substantially impact the accuracy of the findings. Similarly, the time gap between an activity's scheduling and assignment times could also be significantly large for it to substantially impact certain analyses (e.g. activity duration - see Fig. 3).

Engagement with stakeholders to identify correct filtering criteria is crucial. Our initial (naive) filtering of data based on timeframe was not useful. While this filtering resulted in a set of smaller data slices, each slice still contained inherently different processes due to the different behaviours in which claims were processed between *departments*. Thus, our initial findings were 'meaningless' as the results were derived from analysing irrelevant data. As we realized later on, filtering of the log based on department was what the stakeholders expected.

The *order* in which data is filtered is important. For example, assume a filtering rule A (remove all events which were not executed by a resource R) and a filtering rule B (remove all cases that did not complete within 7 days). As shown in Fig. 4, applying A *followed by* B will result in a different log as compared to applying B then A. We found this to be an easy mistake to commit as it is commonly thought that applying rules in an 'and' sequence will result in the same set of outcomes regardless of the order.

XESame is a versatile data filtering tool as it allows the use of rich filtering expressions; however, it is complex, not quite intuitive, and has performance issues for logs with tens of thousands of events. Disco is an intuitive tool and scales well to logs with millions of events. It has powerful filtering capabilities (e.g. based on timeframe and process variants), and it summarizes log information and basic performance statistics concisely (e.g. case duration). Thus, Disco facilitates a quick overview of the processes captured in the log and is also favored by the stakeholders. However, Disco is unable to create an XES/MXML log capturing more than two event types (i.e. 'start' and 'complete') from the
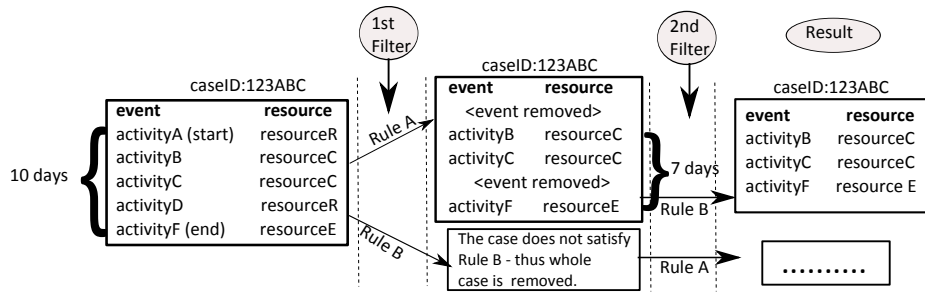
Fig. 4: Illustration showing that order matters when applying filtering rules.

wider set of possible event types involving the lifecycle of a work item. The data used in this case study contained three event types for each work item ('schedule', 'assign', and 'complete'). These three event types cannot be properly translated using Disco, but can be captured using XESame. Disco is also unable to produce an XES log that explicitly distinguishes case-level attributes from event-level attributes (a problem not encountered with XESame). These trade-offs also meant that we often needed to use both tools in our case study.

Result. This stage resulted in a clear identification of slices of data to be used for analysis. The data was firstly split according to the individual departments to which each claim belonged, before being further split into four performance classes based on the two variables captured in the definition of the 'simple' claim: case duration and net payout amount (Fig. 5). The cases in the Simple Quick (SQ) and Complex Slow (CS) groups are expected by definition. The outliers were those cases in the Simple Slow (SS) and Complex Quick (CQ) groups. Cases in the SS class are undesirable outliers while cases in the CQ class are desirable outliers ('complex' cases that were completed rather quickly).
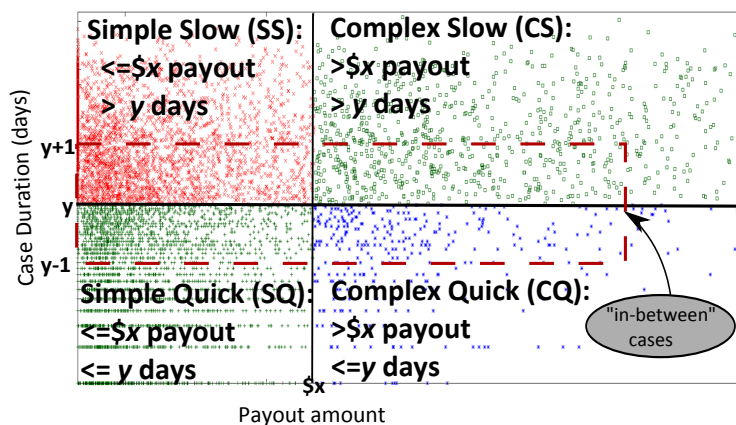

Fig. 5: The slicing of the logs into four performance classes.

3.3 Process Mining Analysis

This stage addressed Q1 (using Disco) and Q2 (using a combination of process discovery algorithms in the ProM Tool and Disco).

Challenges. The main challenge faced during this stage was deriving usable and representative process models from the data. A quick observation of the data showed a very high number of process variants: on average, each variant was executed by at most two cases (out of thousands of cases). Thus, it was anticipated that we would obtain very complex and unreadable process models that would be of limited use. To address this problem, we applied multiple filtering rules one after another (known as 'hierarchical' filtering [15]) to obtain multiple smaller data sets, each of which was likely to produce a simpler and more readable process model.

Similar to the previous stage, this stage was also time consuming and resource-intensive. Certain analysis algorithms, notably Genetic miner [9], ILP miner [22], Heuristic Miner [24], and trace clustering, took a very long time to complete (from a few hours to over a day). Given that not every analysis produced the desired results, similar analyses (with different parameters) would need to be conducted, resulting in a lengthy analysis period.

Lessons Learned. We learned clues for the non-existence of structured processes which include: (1) a very high number of process variants (commonly manifested in a spaghetti-like model) remain despite having applied hierarchical filtering to a reasonable extent, (2) the generation of Heuristic nets with very low *fitness* values (a measure of how well the discovered models can describe the behaviours seen in the log, ranging from 0 to 1, with 1 representing perfect fitness) despite the application of various filtering and clustering techniques, and (3) the tendency for the simplified Petri Nets models to be very flexible (almost like a 'flower' model [19]) when they have high fitness values.

The Heuristic Miner [24] is able to generate understandable models, even from complex event logs. A similar observation holds for the ILP miner [22] and the Fuzzy Miner [4]s. We did not have positive experiences with the Genetic miner - it took a very long time while producing results comparable to other techniques. The model simplification tool Uma [2] is useful in reducing the complexity of the discovered Petri Net models.

Results. Q1 and Q2 were addressed. The claims performance distribution is shown in Fig. 5. Note the significantly-high number of cases which fell in the SS class - a problematic situation that is investigated further in Section 3.4.

The results obtained to address Q2 are interesting. We anticipated structured process models as the claim processes were supported by a workflow-like system. However, our results suggested otherwise: claims were processed in different manners, none of which represents *typical* claims processing sequences. For example, Fig. 6 (left) shows an example of the process model obtained through the application of the Fuzzy miner [4] - a very complex model and of limited use.
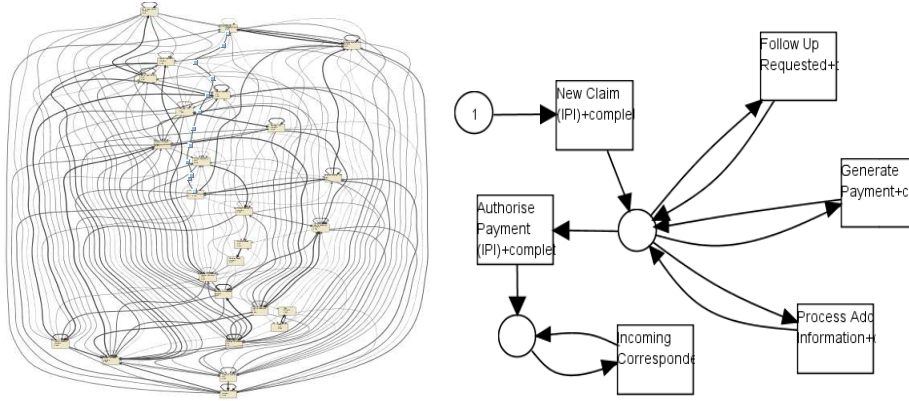
Fig. 6: A spaghetti-like model (left) and a flower-like model (right) discovered

Unfortunately, even after the application of *hierarchical* filtering, the discovered process models were just as complex. Using the Heuristic Miner [24], we obtained a more 'structured' process model (using all events in the data sets), but the fitness values of the discovered models are low (<0.2). We also applied a trace clustering technique [1] (on an already filtered data set) from which 12 clusters of trace variants were obtained. We then used the ILP Miner [22] and Uma [2] to obtain the underlying Petri net models from each cluster. However, the obtained models from all clusters - see Fig. 6 (right) for an example, though of improved fitness (generaly >0.6), allow many extra behaviours. Based on these results, it is likely that there is indeed *no typical* claims processing behaviour. This could be due to the lack of restrictions in the claim processing system (e.g. no prescriptive process models underpinning the system).

Fig. 7 shows the dominant paths discovered using Disco. By generating process models at a higher-level of granularity, the 'dominant paths' could be easily identified as they were clearly marked by thicker lines. These insights proved to be useful for later analysis (see Section 3.4). Certain observations about the process behaviour were quite surprising to the stakeholders (e.g. the unexpected repetition of 'New Claim' activity within a case). This observation warrants further investigation that is outside the scope of this case study.

3.4 Interpretation and Explanation

This stage focused on interpreting and explaining the results obtained from the previous stage. Q3 was addressed in this stage. Using the *process log-based RCA* approach mentioned in Section 2, we enriched the log with relevant *predictor* variables related to process behaviours (e.g. activity execution frequencies) which were not readily available in the log. The enriched log was aggregated into a case log whereby each case was explicitly tagged with its appropriate *response* variable according to the four performance classes defined earlier (see Fig. 5). To address Q3, we applied a number of *classification* analyses (using the WEKA
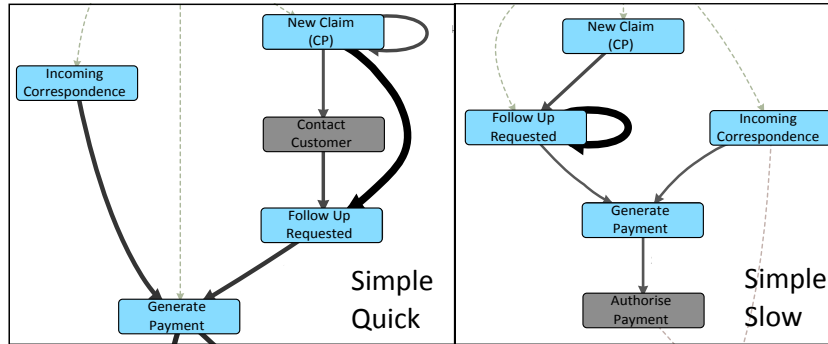
Fig. 7: A snippet of discovered process models using Disco - thicker lines represent the dominant/frequently-traversed paths. Dashed lines represent activities that happened at the start or the end of the process.

tool [25]), to compare SQ and SS classes to discover rules that may explain lengthy processing times in SS.

Challenge. Given the abundance of variables to consider, our earlier effort to identify relevant *predictor* variables was explorative and unstructured leading to poor results. Finding a balance between having highly-precise *predictor* variables and simple classification rules (usable by practitioners) is a difficult task. Often, we obtained rules that were either too fine-grained (thousands of rules with too many 'if-else' conditions) to be usable in practice, or so coarse that they resulted in low accuracy (e.g. a false-positive rate of more than 50%). Spending time to tune the parameters used in the analysis (e.g. limiting the maximum number of rule branches) could, sometimes, overcome this issue.

Lessons Learned. In interpreting the discovered process models, one needs to be mindful of the inherent assumptions. For example, the graphical representation of a fuzzy net (typically derived from the application of the Fuzzy Miner [4]) does not differentiate between the 'eventually followed by' and the 'directly-followed by' relationships of two activities. Thus, it is easy to make the mistake of concluding that a task, say A, is directly followed by another task X in all cases, even though such a relationship is not always true.

It is better to start the process log-based RCA by comparing classes with high 'contrast' (see Fig. 5): there is no obvious gap in the performance of the cases in the SS and SQ classes due to the 'in-between' cases which act as noise that distorts the accuracy of the results. When comparing two classes with a pronounced performance difference (outside the 'in-between' cases), we obtained results with better accuracy. Hence, it may be better to compare two sub-sets of the log with a pronounced performance difference, and then gradually close the gap to observe the changes in the results.

We also identified two process-related metrics which proved to be useful *predictor* variables: (1) the *average per-case execution frequency of an activity-X* (total occurrence of activity-X/number of cases), and (2) the *distribution of an activity-X* over all cases (total number of cases with an activity-X/number of

cases). The greater the difference between both metrics in two classes, the more likely they can be used as *predictor* variables. Henceforth, we used the term `actFreq` and `actDist` to refer to these two metrics respectively.

These two metrics can be easily identified by examining the discovered process models - see Fig. 7. At the same level of granularity, for SQ cases, the self-loop of 'Follow Up Requested' activity is not a dominant path (hence not shown in the model), while this is the case for SS class. Hence, the `actFreq` metric of this activity is likely to be quite different between these classes. Similarly, the activity 'Contact Customer' does not appear in the SS class but it does in the SQ class. Thus, the `actDist` metric is likely to be different too.

Results. Table 1 compares the `actFreq` and `actDist` metrics between the SS and SQ cases. The difference in the `actFreq` is quite pronounced for a number of activities, notably 'Follow Up Requested', 'Incoming Correspondence', and 'Contact Assessor'. The difference in the `actDist` metric is quite pronounced for the 'Contact Assessor' activity. These activities were thus chosen as *predictor* variables in our subsequent analysis, except 'Conduct File Review' which was uninteresting to the stakeholders (merely a routine task triggered periodically).

Table 1: A snippet of interesting `actFreq` and `actDist` metrics

Activity	Simple Quick		Simple Slow	
	actFreq	actDist	actFreq	actDist
Follow Up Requested	1.86	74.4%	5.79	92.3%
Incoming Correspondence	1.75	81.6%	4.27	90.1%
Contact Customer	0.66	46.8%	1.29	63.3%
Contact Assessor	0.11	4.9%	1.36	21.5%
Conduct File Review	2.03	89.8%	6.11	96.9%

Through *classification* analysis, we obtained a set of rules that afford a preliminary insight into the reasons behind lengthy processing times for cases in SS. For example, one of the rules discovered states that if the activity 'Follow Up Requested' is executed more than 4 times, then there is an 85% chance that the process will be lengthy. This rule holds for 55% of all 'Simple Slow' cases. We also found that a claim single-handedly processed by a single resource resulted in a shorter processing time than when it was handled by multiple resources.

3.5 Improvement

The stakeholders have indicated useful (and sometimes surprising) findings that may guide their process improvement actions. For example, insights from Q1 were useful to the stakeholders as they were able to quickly understand the performance distribution of their claims. The high number of cases in the SS class was surprising to the stakeholders as it was more than what they originally thought. This result has therefore highlighted the group of claims that need to be improved. Insights from Q2 have not only validated the stakeholders' suspicion w.r.t the non-standard manner in which their claims were processed, but

also highlighted the need for process standardization. Insights from Q3 have precisely identified areas for improvement (e.g. the flagging of certain problematic activities has precisely identified the pain points in their processes).

Our case study has managed to deliver positive impact to the stakeholders. As stated by an employee of Suncorp, *"..by mining and analysing our ... claims..., our business has been able to make cost saving adjustments to the existing process."*

4 Recommendations

It is useful to apply an appropriate methodology to managing a process mining project. Section 3.1 explained the importance of conducting a 'question-driven' process mining project as suggested by the L^*-methodology. Furthermore, the case study could have been carried out within a shorter timeframe had we followed the methodology more closely. For example, in the early phase of our case study, deviations occurred whereby data extraction took place before the planning stage was properly completed, resulting in a waste of time doing unnecessary explorative analyses without a clear direction as to what questions are interesting to be addressed from the viewpoint of the stakeholders. This situation triggered another round of lengthy data extraction, cleaning, and pre-processing.

Ensure a well-balanced team and maintain a close engagement with stakeholders. As detailed in Sections 3.1 and 3.2, the involvement of stakeholders, a process improvement expert, a process mining expert, and IT personnel are essential for successful and timely completion of the project. Having a close engagement with stakeholders is indispensable to properly interpret the data, to correctly filter the data for analysis, and to avoid wasting time doing unnecessary analyses. For example, without insights from the stakeholders, we would not know the importance of filtering data by *department* to obtain meaningful analyses. We also would have wasted efforts in analysing certain activities (e.g. 'Conduct File Review') which seemed to be significant from a data analysis point of view, but were meaningless to stakeholders as they were routine tasks.

Do spend time to understand the data and to identify research questions. As explained in Sections 3.1 and 3.2, we need to have a set of clear questions to guide the direction of a process mining project, in addition to having sufficient data to address these questions. The data is likely to contain a lot of noise (e.g. a high number of infrequent process variants), which may be due to the existence of multiple different processes in the data. Therefore, the ability to distinguish the various sub-processes in the data, in addition to being clear about the *interpretation* of the data, is crucial for conducting proper data filtering activities. For example, by separating activities involved in an insurance claim into their respective sub-processes (e.g. claim lodgement, assessment, payment, and recovery), we may be able to reduce the number of process variants within each sub-process and simplify the subsequent analyses. Note that not all questions can be answered through process mining analysis; in our case, we complemented process mining analysis with traditional data mining analysis to address Q3.

Choose log filtering tools appropriately according to the size and complexity of the log. In Section 3.2, we have elaborated on the advantages and disadvantages of two log filtering tools: XESame and Disco. To summarize, XESame is a flexible data filtering tool that also allows rich and flexible data manipulation; however, it has performance issues handling large event logs with rich attributes. Disco, while having rich data filtering capabilities and handling large logs well, has its own limitations detailed in Section 3.2.

Plan analysis steps strategically. As explained in Section 3.3, executing a process mining analysis can be time consuming, depending on the nature of the algorithm, the complexity of the log, and the size of the log. One may consider splitting or sampling the data for analysis, according to the availability of time and computing resources. Recent research in the splitting of large process logs for process mining analysis [17] may eventually address the performance issues that we currently face.

Use a combination of algorithms to deal with complex unstructured processes. As detailed in Section 3.3, we found that using the ActiTrac clustering, ILP Miner, Fuzzy Miner, and Heuristic Miner in combination with Uma together can handle complex logs quite well. A useful *sequence of analysis* would be to cluster logs using a trace clustering technique, followed by process model discovery (using one or more of the algorithms listed earlier) from each cluster. Then, convert the best-fitting models (from each cluster) to Petri Nets before applying Uma (if the Petri Nets models are too complex/non-readable). Finally, run conformance check analyses to gauge the fitness of the models.

5 Related Work

This section discusses similar work related to the application of process mining in practice. For general process mining literature (including how it compares to other forms of business analytics), please refer to the work by van der Aalst et al. [15, 16, 19].

Process mining case studies using real-life data have been performed in different domains, e.g. public service [18], manufacturing industry [11], finance [5,12], and health [8,10]. Our case study (with Suncorp) faced similar challenges to those reported in the related work. For example, the problem of noise in the data was also encountered in [5,18]. The problem of "explorative and undirected" analysis encountered in our case study was also found in [13] (which, however, focused on data mining rather than process mining).

Our case study also confirms the validity of some observations made in these related work. For example, the importance of having a close interaction with stakeholders, observed in [18], was highlighted in our case study. The usefulness of the Heuristic Miner in deriving comprehensible models from unstructured processes (an observation made in [8, 11, 18]) was also confirmed in our study. The observation made in [6] about the need to be mindful about the differences in case-level and event-level data was also experienced in our case study.

More importantly, our case study has contributed to a number of additional insights which may benefit other practitioners. These insights include detailed explanations of the challenges and lessons learned at every stage of a process mining project, the importance of team make-up in the project, a performance review of various process mining algorithms, and the strengths and weaknesses of two log filtering tools. Therefore, the current case study can be seen as a significant contribution to improve the adoption of process mining in practice (which is still limited, particularly in the domain of finance in Australia).

6 Conclusion

We have elaborated the challenges, the lessons learned, and the results from the application of process mining techniques in one of the largest insurance companies in Australia (Suncorp). The experiences from our case study have confirmed several challenges and lessons learned in other related studies, and added new insights that can benefit other practitioners. We have also shown some results from this case study which may be used by Suncorp to improve their insurance claims process. Future work may include an in-depth analysis of the resource perspective and the application of a pattern abstraction technique to better understand the processes captured in the event logs from Suncorp.

References

1. J. De Weerdt, S.K.L.M. vanden Broucke, J. Vanthienen, and B. Baesens. Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes. In *IEEE CEC*, pages 1 –8. IEEE, 2012.
2. D. Fahland and W.M.P. van der Aalst. Simplifying mined process models: an approach based on unfoldings. In *BPM*, volume 6896 of *LNCS*, pages 362–378. Springer, 2011.
3. Gartner. Gartner exp worldwide survey of nearly 1,600 CIOs shows IT budgets in 2010 to be at 2005 levels. *Gartner*, 2010.
4. C. Gunther and W.M.P. van der Aalst. Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In *BPM*, volume 4714 of *LNCS*. Springer, 2007.
5. M. Jans, J.M. van der Werf, N. Lybaert, and K. Vanhoof. A business process mining application for internal transaction fraud mitigation. *Expert Syst. Appl.*, 38(10):13351–13359, September 2011.

6. M.J. Jans. Process mining in auditing: From current limitations to future challenges. In *BPM Workshops, Part (II)*, volume 100 of *LNBIP*, pages 394–397. Springer, 2012.

7. T. Jones. Understand how ABPD can help improve business processes. *Gartner*, March 2012.

8. R. Mans et al. Process mining techniques: an application to stroke care. In *Ehealth Beyond the Horizon- Get it There*, pages 573–578. IOS Press, 2008.

9. A.K.A. Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Genetic process mining: an experimental evaluation. *Data Mining and Knowledge Discovery*, 14:245–304, 2007.

10. Á. Rebuge and D.R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Inf. Syst.*, 37(2):99–116, April 2012.

11. A. Rozinat, I.S.M. de Jong, C.W. Gunther, and W.M.P. van der Aalst. Process mining applied to the test process of wafer scanners in ASML. *IEEE Trans. on System., Man, and Cybernetics, Part C*, 39(4):474 –479, 2009.

12. M.S. Saravanan. Application of process mining in insurance: A case study for UTI. *Int. Jour. of Adv. Comp. and Math. Sciences*, 2(3):141–150, 2011.

13. K.A. Smith, R.J. Willis, and M. Brooks. An analysis of customer retention and insurance claim patterns using data mining: a case study. *Journal of the Operational Research Society*, 51(5):532–541, May 2000.

14. S. Suriadi, C. Ouyang, W.M.P. van der Aalst, and A.H.M ter Hofstede. Root cause analysis with enriched process logs. In *BPM 2012 Workshops*, volume 132 of *LNBIP*, pages 174–186. Springer, 2013.

15. W.M.P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

16. W.M.P. van der Aalst. Using process mining to bridge the gap between BI and BPM. *Computer*, 44(12):77–80, December 2011.

17. W.M.P. van der Aalst. Decomposing process mining problems using passages. In *ICATPN*, volume 7347 of *LNCS*, pages 72–91. Springer, 2012.

18. W.M.P. van der Aalst et al. Business process mining: An industrial application. *Inf. Syst.*, 32:713–732, 2007.

19. W.M.P. van der Aalst et al. Process mining manifesto. In *BPM Workshops (1)*, volume 99 of *LNBIP*, pages 169–194. Springer, 2011.

20. W.M.P. van der Aalst, H. A. Reijers, and M. Song. Discovering social networks from event logs. *Comput. Supported Coop. Work*, 14(6):549–593, December 2005.

21. T.H.C. van der Heijden. Process mining project methodology: Developing a general approach to apply process mining in practice. Master's thesis, Eindhoven University of Technology, 2012.

22. T. van der Wiel. Process mining using integer linear programming. Master's thesis, Eindhoven University of Technology, 2010.

23. H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. XES, XESame, and ProM 6. In *Information Systems Evolution*, volume 72 of *LNBIP*, pages 60–75. Springer, 2011.

24. A.J.M.M Weijters, W.M.P. van der Aalst, and A.K.A. Medeiros. Process mining with the heuristic miner-algorithm. BETA Working Paper Series WP 166, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.

25. I.H. Witten et al. Weka: Practical machine learning tools and techniques with Java implementations. In *Emerging Knowledge Engineering and Connectionist-Based Information Systems*, pages 192–196. University of Otago, 1999.