Efficient and Compliant Purchase Order Handling

A Contribution to BPI Challenge 2019

Oliver Gutermuth, Johannes Lahann, Jana-Rebecca Rehse, Martin Scheid, Steffen Schuhmann, Sebastian Stephan, Peter Fettke

Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI GmbH) and Saarland University, Campus D3.2, Saarbrücken, Germany
firstname.lastname@dfki.de

Abstract. The 2019 Business Process Intelligence challenge focuses on the purchase order handling process from a large multinational company in the area of paints and coatings, operating from the Netherlands. In this report, we describe how we analyzed the provided process data in order to answer the process owner's questions on process visualization, throughput efficiency, and compliance. To answer those questions, we used a combination of manual data analysis, established process mining techniques, and innovative machine learning approaches. After presenting our data understanding and tool chain, we first report on the results obtained by manual filtering, before addressing each challenge individually. We also discuss limitations and further recommendations, wherever applicable.

Keywords: Process Mining, Process Discovery, Process Compliance, Anomaly Detection, BPI Challenge

1 Introduction

With the help of information systems, business processes are increasingly digitized. Through the (semi-)automatic execution of process activities, the performance and the quality of the process can be improved. However, the proceeding digitization of business processes can also lead to an increasing complexity of organizations. This is particularly challenging in the context of ensuring process compliance, i.e., conformance with internal and external regulations. In this context, the recorded process data can be used to gain insights into the process, as well as its accompanying data, organizational, and social structures, with the help of process mining.

The yearly BPI challenge (BPIC) gives process mining researchers and professionals to test their tools, techniques, and methods on a real-life log. For the BPI Challenge 2019, the provided data comes from a large multinational company in the area of coatings and paints operating from The Netherlands. In particular, it is focused on the purchase order handling process for some of its 60 subsidiaries. This process is concerned with administrating and documenting purchases, including goods receipt and invoices. Each case corresponds to one purchase order item (or line item). Purchase orders (or

purchase documents) may contain one or more purchase order items. For each line item, there are four different ways to handle it. If an item requires both an invoice and a goods receipt, the invoice may either be issued independent from the goods receipt (3-way-matching, invoice before goods receipt) or only after the goods receipt is entered (3-way-matching, invoice after goods receipt). Other items require an invoice without a goods receipt (2-way-matching) or vice versa (consignment).

From an analytical perspective, the process is challenging, because both the data and the underlying process are quite complex. This complexity not only makes it difficult the optimize the overall process flow, it also complicates to ensure its compliance regarding internal and external regulations. Process mining, artificial intelligence, and data analytics may help to reduce this complexity. According to the problem statement on the BPIC 2019 website, the providing company is interested in answering the following concrete questions:

- Is there a collection of process models which together properly describe the process in this data? Based on the four categories above, at least four models are needed, but any collection of models that together explain the process well is appreciated. Preferably, the decision which model explains which purchase item best is based on properties of the item.
- What is the throughput of the invoicing process, i.e. the time between goods receipt, invoice receipt and payment (clear invoice)? To answer this, a technique is sought to match these events within a line item, i.e. if there are multiple goods receipt messages and multiple invoices within a line item, how are they related and which belong together?
- Finally, which Purchase Documents stand out from the log? Where are deviations from the processes discovered in (1) and how severe are these deviations? Deviations may be according to the prescribed high-level process flow, but also with respect to the values of the invoices. Which customers produce a lot of rework as invoices are wrong, etc.?

To answer these questions, this report is organized as follows. In Sect. 2, we perform a first data analysis, describing which tools we have used, our data understanding and the results of a descriptive analysis. Manual data filtering is done in Sect. 3. Sect. 4 is centered on the first challenge, describing how the process log could be categorized further in order to simplify its analysis. In Sect. 5, we address the second challenge, using a combination of manual analytic tasks and automated data filtering. Section 6 is dedicated to the last challenge, identifying abnormal cases based on the results from the previous challenges, before the paper is concluded in Sect. 7.

2 Preliminary Analysis

2.1 Software Tools

A variety of tools was used to analyze the data from different perspectives and providing answers. While established tools were mainly used for descriptive analytics and

standard process mining tasks, individualized solutions were implemented for the more challenging and data-specific questions. The following table provides a summary of the software frameworks and tools that have been used for generating the results presented in this report.

Table 1: Software tools and frameworks used for data analysis

Framework / Tool	Purpose
Fluxicon Disco	Process mining, process discovery, descriptive analysis
ProM 6.8	Process mining, process discovery, visualization
RefMod-Miner	Log manipulation, log conversion, trace similarity
PM4Py	Token Replay
Python 3.7 Including pandas and numpy	Log manipulation, descriptive analysis, clustering, anomaly detection
Pytorch	Language model creation
FastAI	Language model creation
Matplotlib	Visualizing
MS Excel	Analyzing a selection of individual logs using Power Pivot

2.2 Process Understanding

The main goal of the BPIC is to inspect the purchase order handling process regarding compliance issues. For this purpose, the purchase orders including the individual purchase order items should be analyzed. According to the BPIC 2019 website, purchase order items themselves can be processed using the following four different procedures, as summarized in Table 2. Here, a Product Order (PO) refers to each individual purchase order item, an invoice (IV) denotes the vendor invoice (not the invoicing) and goods receipt (GR) stands for the goods receipt document, e.g., a delivery note.

Table 2: Overview of the item categories

	3-way matching, invoice after GR	**3-way matching, invoice before GR**	**2-way matching**	**Consignment**
GR-flag	TRUE	TRUE	FALSE	TRUE
GR-based-IV-flag	TRUE	FALSE	FALSE	FALSE
Verification	PO + IV + GR	PO + IV + GR	PO + IV	PO + GR
Special feature	Invoice may only be entered after GR	Invoice can be entered before GR; however, settlement only takes place after GR	Settlement is independent of GR	Billing takes place in a separate process

As a further note, it was specified that several goods receipt declarations and invoices can be recorded for individual items (e.g., for installment payments or orders for time-delayed batches). This makes it difficult to compare the amounts, whose sum must match in the end. The values of each event were anonymized linearly such that their relation is preserved. In addition, the approval procedures (for purchase orders and settlement of invoices) were removed.

We also distinguish between two types of users: automated processes ("batch users") and human actors ("normal users"). However, not all events are associated with a specified user. Finally, it should be mentioned that company, supplier, system, document name, and IDs have been anonymized, but product details are not.

2.3 Data Description

Before we start to discover and analyze process models, it is important to establish a general data understanding. This helps us to identify outliers and later filter the event log. For this BPIC, we will focus on the most common KPIs provided by DISCO. Using the project file provided by Fluxicon, we filtered the event log regarding the case Item category and compare the results with the whole event log.

Table 3: Data description along the four item categories

KPI	3-way matching, invoice after GR	3-way matching, invoice before GR	2-way matching	Consignment	Overall
#Events	319,233	1,234,708	5,898	36,084	1,595,923
#Start-Events	5	8	4	2	8
#End-Events	24	28	8	11	32
#Activities	38	39	11	15	42
#Cases	15,182	221,010	1,044	14,498	251,734
#Variants	4,228	7,832	148	281	11.973
Earliest Event	23.01.2001 23:59:00	26.01.1948 23:59:00	25.01.2017 23:59:00	31.12.2017 00:00:00	26.01.1948 23:59:00
Latest Event	09.04.2020 23:59:00	05.12.2019 23:59:00	01.02.2019 23:59:00	17.01.2019 22:26:00	09.04.2020 23:59:00
Median Case Duration	63.4 days	66.3 days	23.7 days	19.9 days	64 days
Average Case Duration	75 days	74.5 days	57.5 days	24.1 days	71.5 days
Max. Duration	17 years, 362 days	70 years, 120 days	1 year, 216 days	229 days, 20 hours	70 years, 120 days
Min. Duration	0 millis	0 millis	2 mins	0 millis	0 millis

2.4 Descriptive process analysis

As a next step, we employ process discovery to gain insight into the as-is process flow. Most commercial process mining tools (e.g. Fluxicon, Celonis) rely on heuristics miner and fuzzy miner. While heuristic process discovery algorithms are widely used and well-established. , we use the interactive Data-aware Heuristics Miner (iDHM) [1] for descriptive analysis in the following. The interactive exploration of the parameter space and the quality of the mined process model will provide a broad process understanding as a prerequisite for advanced analyses.

The iDHM provides several mining algorithms. Having a first look at the process model, we use the Flexible Heuristics Miner (FHM) [2]. Furthermore, we will use the all-task-connected heuristic. As stated in [1, 2], the advantage of using this heuristic is that many dependency relations are tracked without any influence of the used parameter setting.

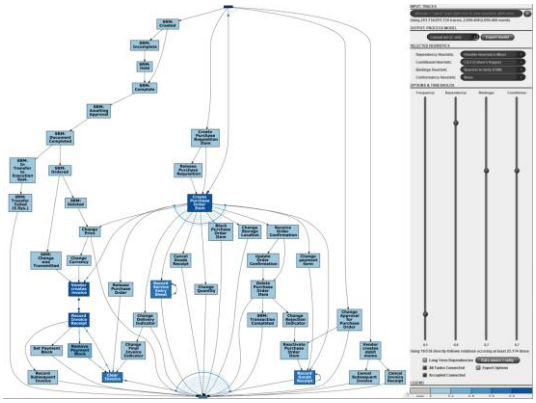

Figure 1: Mined process model with FHM (event log)

The heuristics calculates dependencies between events in a so-called dependency graph. Because the metric indicates how there is truly a dependency relation between two activities [3, 4], we only look at traces with a dependency >= 0.9. As depicted in Figure 1, several dependencies can be derived: (i) In general, the process starts with the creation of a purchase requisition item or the creation of a purchase order item. (ii) The activity *Record Service Entry Sheet* is only executed together with *Record Goods Receipt*. (iii) The activities *Vendor create invoice* and *Record Goods Receipt* are executed together. However, in some cases no vendor invoice is created.

In a next step, we compare the discovered process models regarding the four flow types. Therefore, we filtered the given event log regarding the Case Item Category to get four event logs. Doing this, all four flows look different. The bindings next to the artificial start event represent a parallel split, whereas disconnected dots represent exclusive splits. Looking at the process models, we can observe that for the 2-way matching, no goods receipt is needed (**Figure 2**). Analyzing the 2-way matching model in more detail, we see that after the invoice receipt is recorded, there are three possible ways to end the process. Conspicuously, there are cases for which no activity *clear invoice* can be observed. In addition, canceling the invoice receipt is contradictory to paying the invoice afterwards.

For the consignment flow, several activities which lead to *Record Goods Receipt* can be observed. As it is depicted in **Figure 3**, different actions can happen before the activity *Record Goods Receipt* is executed. It is interesting that there is a parallel split before the activities *Cancel Goods Receipt* and *Receive Order Confirmation*. It also looks like that in order to delete a purchase order item, an order confirmation has to be received. Here, a point of interest could be to analyze when a purchase order item is

reactivated, and when not. In general, we see that all updates and changes regarding goods receipt happens before they are recorded. For the consignment process, it looks like that there are no cases where a recorded goods receipt is updated afterwards.

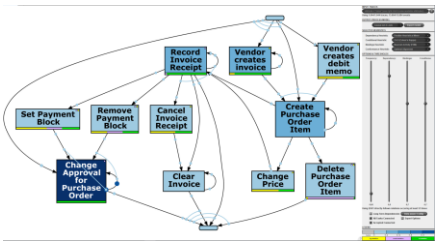

Figure 2: Discovered 2-way matching process (FHM)

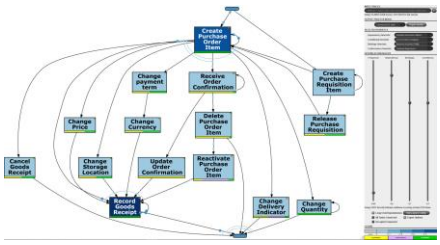

Figure 3: Discovered consignment process (FHM)

Looking at Figure 4, the flow for 3-way matching (invoice after GR) is much more complex. If we analyze the start point, we see that only 1,221 traces start with *Release Purchase Order* (1) or *Create Purchase Requisition Item* (1.220). According to the bindings, all other traces starts with more than one activity (parallel split). For example, we see that 7,440 traces start with the activity *Record Goods Receipt* AND *Create Purchase Order Item*. Interestingly, there are traces which consist of the activities *Vendor creates debit memo* and *Cancel Invoice Receipt*. These traces are conspicuous because no activity *Record Invoice Receipt* is executed.

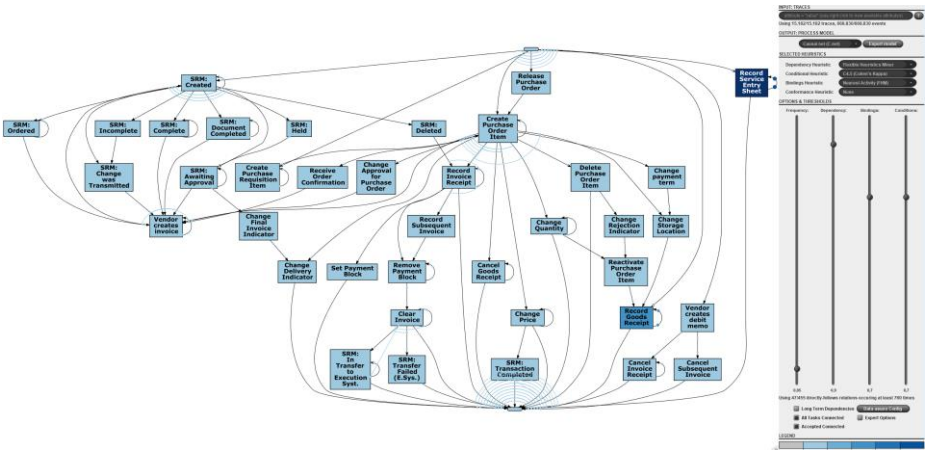

Figure 4: Discovered 3-way matching (invoice after GR) process (FHM)

If we look at Figure 5 (3-way matching, invoice before GR), we can observe that there are more or less the same SRM activities as in the process model shown before. In addition, looking at the structure, it looks like the SRM-activities form a separate and internally cohesive sub-process. As this cannot be observed in the previous two models, these activities are particular for 3-way matching flows.

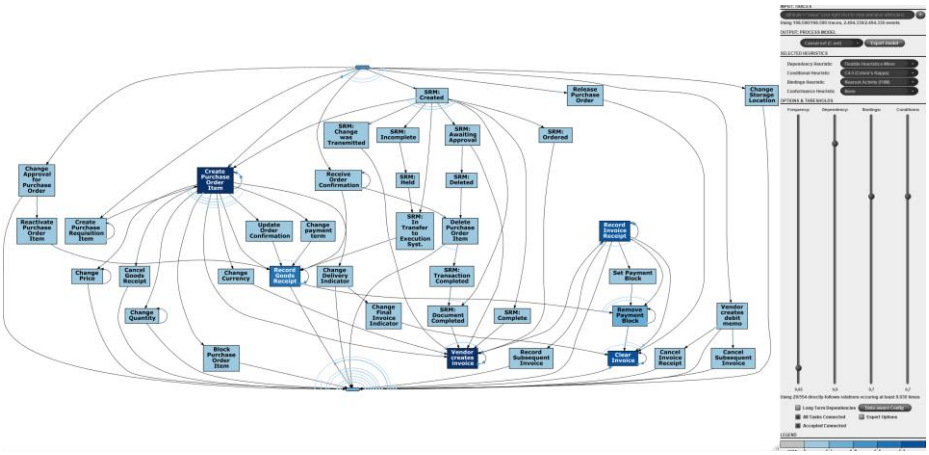

Figure 5: Discovered 3-way matching (invoice before GR) process (FHM)

3 Data Filtering

3.1 Approach

In a first step, the log was divided by process type (consignment, 2-way-matching, 3-way-matching with invoice before and after GR). This results in four sub-process logs. These sub-logs were then imported into Disco and analyzed. First, we focused on start events, beginning by inspecting the start events and decided whether or not to filter these cases according to the given timestamps. We also looked at some additional attributes of the log to find cases with potential start events before or after 2018. Additionally, we looked at other attributes, such as the order of events and their frequency to decide whether or not those cases will be included. Next, we followed an analogous procedure for the end events of the log. We looked at all potential end events and decided whether or not the corresponding cases will be included. In addition, if we were unclear about the sequence or logic behind the events, we looked at SAP's help (*https://help.sap.com* or *https://answers.sap.com*) to get a deeper understanding of the process and activity logic. This procedure was carried out on all four sub logs of the BPI Challenge dataset. Manually filtered results are also described as potential process anomalies in Sect. 6.2. The detailed filtering tables are available in Appendix A. The models resulting from this process can be considered as a proposal for the definition of a manually created reference process.

3.2 Findings of the manual filtering process of the consignment sub log

First, we filtered for the consignment process type which comprises 14,498 out of 251,734 cases. We looked for possible start events, i.e., events which occur as the first in time within a case. We identified six potential start activities (see Appendix A **Table 11**). Looking at the occurrence of the events inside the consignment sub log shows that

three of the activities are start events which resulted from logs which started before 2018 and therefore resulted from cases which started before the time period we are looking at. These are therefore probably incomplete cases, where the real beginning was cut off. So, we did not include those cases inside our model. The activities *Delete Purchase Order Item* only functions as start event in cases which consists of itself and *Create Purchase Order Item*, so these are also filtered. This results in two start events: *Create Purchase Requisition Item* and *Create Purchase Order Item.*

Next, we inspected potential end events (see Appendix A **Table 12**). When considering these activities and the instantiated events, it became clear that eight of them were end events of pending process instances and were therefore no longer taken into account as end events. *Create Purchase Order Item* is only instantiated as end event if it is also start event in December 2018. We also found that *Delete Purchase Order Item* is only an end event if some process attribute was changed and update, cancel, change or reactivate events took place before. This resulted in the end events *Delete Purchase Order Item* and *Records Goods Receipt.* This filtering procedure subsequently led to a log with 13,534 cases and was subsequently discovered using DISCO (see Appendix B **Figure 21**). We also followed those steps in the following chapters.

3.3 Findings of the manual filtering process of the 2-way-matching sub log

The 2-way-matching sub log consists of only 1,044 cases and contains five potential start events (see Appendix A **Table 13**). *The filtering then resulted in the start events Create purchase order item* and *Vendor creates invoice.* Next, we identified potential end events based on the already found start events, finding seven potential end events (see Appendix A **Table 14**). This leads to *Clear Invoice* and *Records Invoice Receipt* as end events. Using those endpoint filters in addition with the filtered timeframe led to the discovered model in Appendix B Figure 22, based on 1,044 cases.

3.4 Findings of the manual filtering process of the 3-way-matching after GR sub log

In this subprocess, we first extract the SRM subprocess. The SRM activities appear very closely connected and are probably carried out conjointly in Supplier Relationship Management, a separate SAP module. Therefore, we filter this process out of the log and analyze it separately (538 out of 251,734). This led to the model visualized in Appendix B Figure 24. In the following, the activities considered here were filtered out in order to get a more exact impression of the remaining 3-way-matching log. The first filter results of the potential start events are listed in Appendix A Table 15. This results in a large number of potential end events. After the detailed analysis of these events we found eight potential end events, listed in Appendix A Table 16. For these cases, however, no exact decision could be made, so they were not filtered. The other activities resulted from pending and timeframe overlapping process instances. After filtering for those start and end points, we discovered the model visualized in Figure 6. However, it must be considered that due to the reduction of the displayed paths, some start and end

events are not directly recognizable as such, since they are only present as such in a few cases.

3.5 Findings of the manual filtering process of the 3-way-matching before GR sub log

After filtering for the 3-way-match before GR sub log (194,288 cases), we identified four potential start events. Here, only *Vendor creates debit memo* can be excluded as a start event because of hanging or time-overlapping process instances. This leads to a large number of potential events as process start points (see Appendix A **Table 17**). After further filtering and detailed analysis, we identified e.g. *Cancel Goods Receipt, Cancel Invoice Receipt and Cancel Subsequent Invoice* as end events (see Appendix A **Table 18**). This results in a filtered log which consists of 194,228 cases and is visualized in Figure 7.

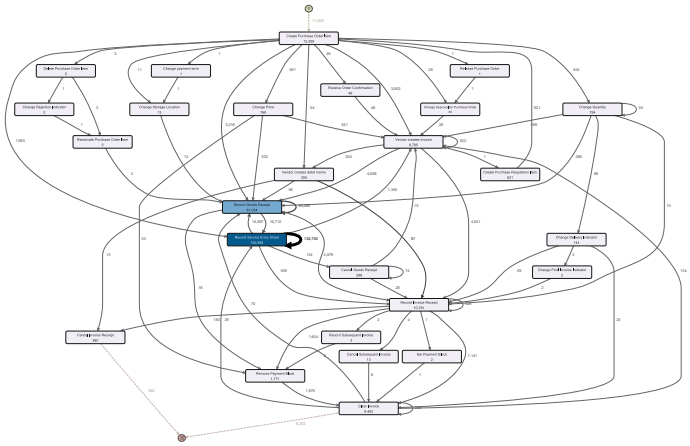

Figure 6: 3-way-matching after GR (100% of the activities and 40% of the paths)

4 Challenge 1: Process Model Collection Mining

4.1 Approach

The process owners' first question was to find a collection of process models, which together properly describe the process in this data. Since there are four fundamental ways to handle a line item, at least four models are needed, but the process owners were open to more models as long as they explain the process well. They preferred a collection of models, where the assignment of line items to models is based on properties of the item.

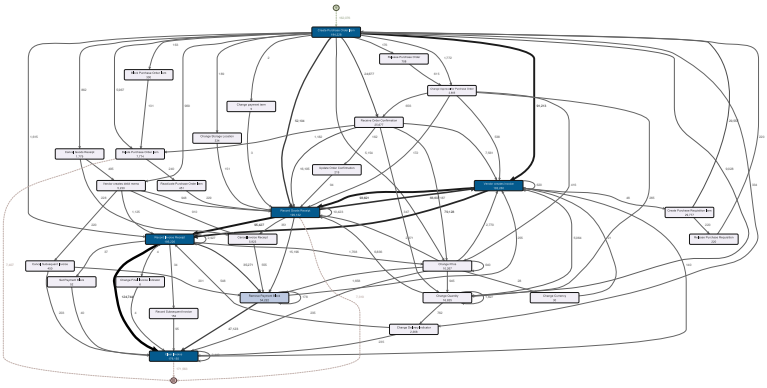

Figure 7: 3-way-matching before GR (100% of the activities and 20% of the paths)

The premise of this challenge was that we should inspect each of the four process types individually, deciding which one could benefit from being split into multiple processes. So, we first inspected the four sub-logs, as described in the previous section 3, but unfiltered to avoid potential distortions. Both the consignment and 2-way matching process that were mined for the respective logs were rather small and structured, giving stakeholders a good understanding of the process flow. For those two subprocesses, we saw no need for further separation. The same, however, could not be said about the two 3-way matching sub-logs. 3-way matching, invoice before GR is the largest log by a considerable margin; its many variants make it quite difficult to identify a clear structure. The same can be said about 3-way matching, invoice after GR, which is remarkable, because this log is much smaller. Nevertheless, manual filtering revealed that it has a very high number of distinct process variants. So, in this challenge, we focused on the two 3-way matching sub-logs, with the goal to separate them into more detailed process models. As explained above, we also removed the SRM process from both logs, because we considered it as a separate and independent subprocess of both.

For both datasets, we first approached this challenge in a data-centric way. The basic idea is to use clustering techniques in order to see similarities and differences between groups of process instances within one process type and use those results for ensuing manual analysis. The idea behind clustering techniques is to group a large dataset, such that the datapoints in one group (or cluster) are more similar to each other than to those objects in other groups. The similarity measure between datapoints can be individually defined. For our dataset, we executed the following steps in order to find a viable set of models to describe the purchase order handling process:

1. Separate logs: We separated the overall log into four (complete and unfiltered) sub-logs, each corresponding to one process type. The following steps were executed separately for each log.
2. Compute trace similarity: Trace clustering requires a similarity measure between traces, which can be defined in many different ways [5]. The result is a matrix with

pairwise similarity values for all traces, which was used as the input data for the clustering algorithm.
3. Dimensionality reduction: In order to group similar traces based on the generated similarity matrix, T-SNE was used to reduce the similarity to a two-dimensional space.
4. Separate cluster logs: After dimensionality reduction, our log was separated into multiple groups of similar traces, using the K-means algorithm to identify near located cluster in the projected space, which were then exported as individual logs for further handling.
5. Manual analysis and evaluation: Using Disco and the Inductive Miner [6], the separated logs and mined models then served as the basis for further manual data analysis, which was necessary to find a reasonable number of process models, with a clear connection to line item properties.

4.2 Identifying Process Models for 3-way-matching, invoice before GR

For analyzing the 3-way-matching, invoice before GR sub log, we used a trace clustering based on activity feature vectors. The log is quite large, so we required a computationally efficient way to determine the similarity between two traces. In this approach, the log is represented as a matrix, with a row for each activity and a column for each trace. Each value denotes how often the respective activity appears in the respective matrix. The dimensionality of this feature matrix is then reduced using the T-SNE algorithm, preserving the distance between the datapoints in the projected two-dimensional space. In order to optimize the hyperparameter, we have implemented an iterative approach to use different parameters for perplexity as well as iterations. The results were then compared to find the best hyperparameter combination based on the shape of the resulting clusters. For the further analyses we have chosen a value of 10 for perplexity with 500 iterations.

The results for the 3-way-before log are shown in Figure 8. The most discernible feature is the large cluster centered around the origin. It is surrounded by a ring of more scattered objects. The question remains, how the trace similarity related to features in the log and whether these results can be transformed into a viable set of process models.

After starting to inspect the center cluster, we realized that it corresponded to one specific process variant covering 79,487 cases, more than a third of the log. It consists of five activities (*create purchase order item*, *vendor creates invoice*, *record goods receipt*, *record invoice receipt*, *clear invoice*) and describes the handling of a line item (one case) with exactly one invoice, one goods receipt, and no other features. In the next step, we inspected further process variants, for which the same properties apply (activities *create purchase order item*, *vendor creates invoice*, *record goods receipt*, and *record invoice receipt* appear exactly once). We found that these amount to more than 90% of the 3-way-before log, while still producing a fairly structured process model. Most of the traces were rather small and they all contained each activity exactly once, whereas all other traces repeated activities for multiple invoices or goods receipt. The discovered model is shown in Appendix B Figure 26. The “happy path”, i.e., the most frequently executed process flow is clearly visible, but it also includes the multiple

variants to the process, such as attribute changes, the approval subprocess, or purchase cancellation.

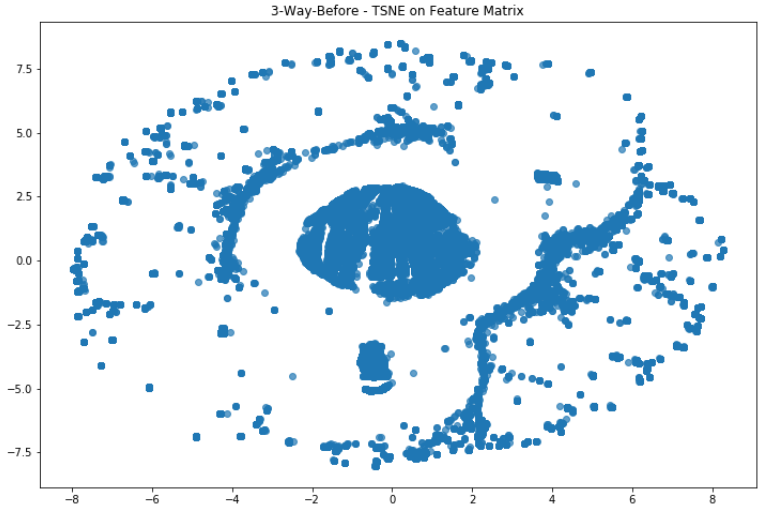

Figure 8: Clustering results for 3-way-before

After separating the line items with one invoice and one good receipt from the log, only about 10% traces remained, covering line items with multiple invoices and goods receipt. The model discovered for these cases is shown in Appendix B Figure 25. While not as structured as the model above, it also shows the generic process structure, including the prevailing repetition of activities.

4.3 Identifying Process Models for 3-way-matching, invoice after GR

Next, we focused on the sub-log for 3-way-matching, invoice after goods receipt. We essentially employed the same procedure as above, but with a different similarity measure. As the log is fairly small (14.571 cases), with 25 activities, but more than 4,700 process variants, our goal was to determine trace similarity with a focus on process structure instead of solely activities. Therefore, we used the Levenshtein trace similarity, defined as the normalized Levenshtein distance between two traces, i.e., the number of insert, delete, and replace operations required to transform one trace into the other [7]. This produced a similarity feature vector for each trace, which was again projected onto a two-dimensional space using the t-SNE algorithm. The clustering results are shown in Figure 9. Again, we see a large cluster centered around the origin and accompanied by a slightly smaller, but also quite compact cluster on its bottom left. Those two are surrounded by a ring of smaller clusters and individual objects.

These results again were the starting point for our manual analysis. Inspecting the individual clusters, we quickly saw that one of the central activities was *record service*

entry sheet, indicating that the corresponding line item was some kind of service. Depending on the cluster, this activity was repeated frequently, typically alternating with *record goods receipt* and leading to many slightly different process variants, depending on the number of times a service was delivered. As the presence of this activity introduced a high variability into the process, we decided to split the log into those traces that contained the activity (presumably service line items) and those that did not (presumably non-service line items).

The process model for the non-service line items was discovered from 9,377 traces and is shown in Appendix B Figure 27. Its structure is similar to the process shown in Appendix B Figure 26, with a clearly discernible "happy path" and several optional or less frequent activities, mostly attribute changes and cancellations. The payment subprocess is isolated and can be found towards the end. Appendix B Figure 23 shows the process model for the service line items, discovered from 5,194 cases. It is less structured than the other processes, but also contains fewer activities (15), which still makes it fairly easy to understand. Even with 0% paths, one can see the frequent loop of activities create purchase order item, record service entry sheet, and record goods receipt, which only becomes more pronounced when displaying more paths.

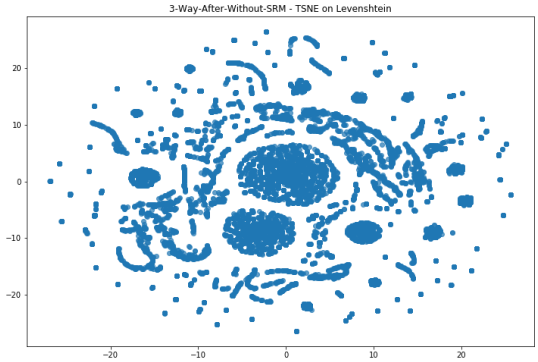

Figure 9: Clustering results for 3-way-after

4.4 Conclusion

To conclude, we suggest describing the process at hand with seven process models:

1. Consignment as shown in Appendix B Figure 21
2. 2-way-match as shown in Appendix B Figure 22
3. 3-way-matching, invoice before goods receipt: items with one invoice and one goods receipt (without SRM subprocess) as shown in Appendix B Figure 26
4. 3-way-matching, invoice before goods receipt: items with multiple invoices and goods receipts (without SRM subprocess) as shown in Appendix B Figure 25
5. 3-way-matching, invoice after goods receipt: service items (without SRM subprocess) as shown in Appendix B Figure 23
6. 3-way-matching, invoice after goods receipt: non-service items (without SRM subprocess) as shown in Appendix B Figure 27

7. Supplier Relationship Management (subprocess) as shown in Appendix B Figure 24

In our opinion, this collection of process models balances out the competing requirements of a having low number of models on the one hand and a having well structured, easy to understand models on the other hand. One could criticize that our separate sublogs differ considerably in size, but they still produce process models of comparable size and structure. This collection also fulfills the requirement that the assignment of line items to models is based on properties of the item. Regarding compliance (as addressed in the next section), it has another advantage: If the matching of line items with their respective invoices and goods receipt is determined by the assigned process model, it becomes a lot easier to check the compliance of individual line items and find potential violations.

5 Challenge 2: Position Matching

5.1 Approach

The second task focusses on the performance of the invoicing process. This process and its variants are important as they are subject to compliance guidelines and related to the four designated flow types. Invoices must be processed efficiently and with a low chance of errors. In addition to compliance requirements, the process steps and their timing should also be inspected, in order to obtain indications of possible disruptions. Analyzing the invoicing process is complicated by the differing complexity of cases within a purchase order document, which may have overlaps or dependencies as well as faulty or incomplete logs. Basically, cases can be defined at the level of a line item or a purchase order. As the selection of compliance procedure uses the properties of an individual item, a categorization on this hierarchical level seems appropriate. The results from Sect. 4 show significant differences between the process flows, depending the frequency of events within a case. Based on this insight, forming two disjunctive sets of cases seemed reasonable.

- Set 1 contains cases having a cardinality of 1:1 between case purchasing document and case concept name
- Set 2 contains cases having a cardinality of 1:n between case purchasing document and case concept: name.

These sets are used for an individual approach to assess the throughput of the invoicing process. To calculate the throughput, events must be selected to define relevant intervals. As there are 42 event types, for which their relevance must be decided, there are two major aspects to consider. The event type label indicating the process flow was used in conjunction with the absolute frequency of its occurrence for a selection. This way, the three most important event types related to the invoicing process were identified (events *Record Goods Receipt, Record Invoice receipt* and *Clear Invoice*) and used for further execution. The main challenge is the systematic examination of all cases containing relevant events of the invoicing process. Accordingly, the occurrence of the three event types per case is determined and the duration between them recorded. Cases

with less than two of them are not considered for the measurement of time intervals. Figure 10 shows a scenario, where every relevant event type has a unique appearance:

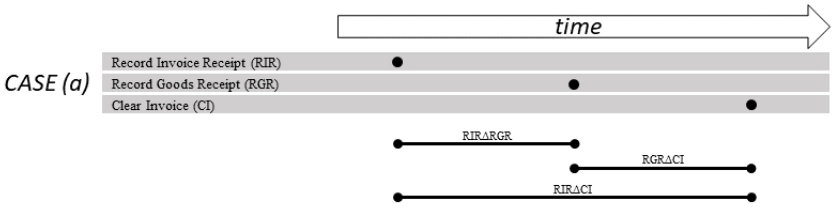

Figure 10: invoicing example: events, sequence, duration

As the order of the appearing events may differ, there are six possible durations to measure, but not all cases can be handled the same way. Basically, the net worth of invoices, the goods receipts, and the clearings should match for a line-item. But as there could be several recurring events for these types within a specific order, the selected method must fit the case characteristics. Therefore, two different methods must be used and chosen considering the cases' complexity. Appropriate approaches must take the frequency of occurring instances of an event type within a single case into account. This ensures that different categories of event quantities within a single event type can be handled separate. Figure 11 shows the criteria determining the procedure.

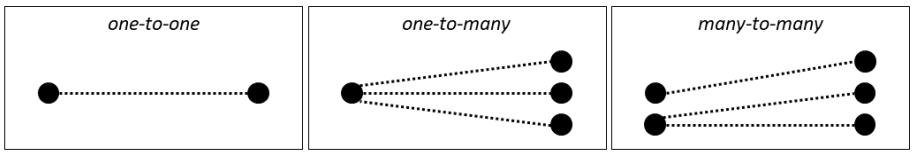

Figure 11: Event quantities and relations

There are three major aspects to consider for measuring the process performance. First, the two sets of cases were built as a significant difference can be expected. Second, the complexity of the relations towards relevant events of the invoicing process leads to the need for two different methods. For one-to-one and one-to-many-relations the explicitness of the sequence allows an accurate measurement of durations as temporal ties are explicit. But the many-to many-relations present the problem of assigning subsequences. A reference was needed. The examination of the XES-file did not provide an appropriate solution as it is limited to values which cannot assign the actual sequence of the process flow when many-to-many-relations appear. The approach to match information from the attribute *event Cumulative net worth (EUR)* failed as too many incongruities were identified. Even the inclusion of the events *Change Quantity, Change Price, Delete Purchase Order Item, Cancel invoice Receipt, Cancel Goods Receipt* which are expected to determine the cumulative net worth could not contribute to the clarification. Thus, the cases containing many-to-many relations were analyzed by using an approximation method considering all events without guessing the actual flow. In this case, the weighted mean of timestamps was calculated for every occurring event

type. The number of relations was derived from the frequency of events within a many-to-many relation. It was assumed that sequences between them occurred just as often as the event with the most frequent expressions per type. For the calculation of average through times the following steps have been processed.

1. Two sets of cases are formed (single line item orders and multiple line item orders).
2. Within each group, all cases containing at least two of the relevant event types with only one type with more than one instance are located, thus analyzing events defined by one-to-one-relations and one-to-many-relations).
3. Depending on the order of occurring events, the average duration between the event types is measured using the accurate method.
4. The cases with many-to-many relations are measured using the approximate method.
5. As the throughput could also address additional useful metrics, the frequencies of event types and the cumulated net worth passing the flow are analyzed.
6. The results are presented and discussed.

5.2 Results

In order to the approach's description the tables within this chapter are created respecting the following scheme.

Table 4: Method dataset combination

	accurate method	approximate method
set-1 (single-case-orders)	**Table 5**	**Table 7**
set-2 (multiple-case-orders	**Table 6**	**Table 8**

To simplify the table's labeling the duration for a measured interval is defined by the timestamp of the limiting events using *RGR* for *record goods receipt*, *RIR* for *record invoice receipt* and *CI* for *clear invoice*.

Table 5: Applying the accurate method on case-set-1 (single-case-orders)

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	7d, 10h	-	-	-
3-way match, invoice after GR	40d, 13h	5d, 23h	33d, 22h	49d, 9h	78d, 16h	-
3-way match, invoice before GR	25d, 0h	4d, 8h	52d, 14h	47d, 23h	67d, 18h	42d, 13h

Table 6: Applying the accurate method on case-set-2 (multiple-case-orders)

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	10d, 1h	-	-	-
3-way match, invoice after GR	38d, 17h	16d, 10h	33d, 11h	41d, 3h	65d, 23h	-
3-way match, invoice before GR	21d, 3h	3d, 6h	48d, 18h	54d, 23h	65d, 17h	15d, 5h

Table 7: Applying the approximate method on case-set-1 (single-case-orders)

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	0d, 18h	-	-	-
3-way match, invoice after GR	28d, 2h	7d, 17h	16d, 2h	9d, 13h	36d, 2h	11d, 12h
3-way match, invoice before GR	27d, 18h	8d, 12h	38d, 2h	47d, 13h	66d, 12h	2d, 4h

Table 8: Applying the approximate method on case-set-2 (multiple-case-orders)

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	9d, 16h	-	-	-
3-way match, invoice after GR	37d, 4h	8d, 5h	27d, 4h	14d, 2h	69d, 4h	10d, 22h
3-way match, invoice before GR	21d, 21h	3d, 11h	48d, 14h	49d, 15h	66d, 0h	15d, 14h

Table 9: Overall average durations

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	311	-	-	-
3-way match, invoice after GR	49026	774	15637	1882	44398	930
3-way match, invoice before GR	203399	16803	182474	1175	191521	85

Table 10: Interval frequencies

Category	Δ(RGR,RIR)	Δ(RIR,RGR)	Δ(RIR,CI)	Δ(CI,RIR)	Δ(RGR,CI)	Δ(CI,RGR)
2-way match	-	-	10d, 18h	-	-	-
3-way match, invoice after GR	24d, 23h	10d, 8h	20d, 9h	16d, 2h	41d, 10h	10d, 11h
3-way match, invoice before GR	23d, 12h	3d, 10h	36d, 11h	36d, 17h	64d, 6h	-

5.3 Conclusion

For evaluating the process performance of invoicing, 708,415 intervals between relevant event types were used. Especially determining their durations and comparing them with respect to the flow variants revealed several findings. On the one hand, for the major part of 653,538 relations accurate key metrics could be calculated. On the other hand, it became obvious how versatile the logs for this process can show off and what kind of challenges their analysis entails. Regardless of the problems occurred by trying to measure complex process flows with many-to-many relations it was possible to also ascertain results for these 54,877 relations.

Not all cases in the data set have been useful for investigating the invoice process. Several of them did not contain any relevant events or they only revealed instances of one single event type, so an interval within the invoicing process could not be identified. This applies to the consignment process, because of the lack of invoices on purchase order level. A closer look at the remaining flow variants offered certain insights. As

indicated by the tables, 2-way match processes do only show off one interval type as GR-events are obsolete and not recognized. Their duration met the expectation that they were rather short compared to other flow variants. However, the duration of the interval between recording invoices and recording goods receipts was recorded with an average of 3 days. A possible reason could be the fact that both events are triggered together as a batch process. The data within the 3-way-procedure confirms the assumption of longer processing times for invoice-after-goods-receipt-flow. The existence of sequences where recording and clearing invoices occur earlier than the record-goods-receipt-events for 3-way-matching with invoice-after-goods-receipt-requirement are remarkable as they show a possible compliance issue within 1704 sequences.

Overall, the results provide an overview of the invoicing process and especially mark the different duration as well as its relation to the respective flow variant. The average durations should be quite accurate as 92.3% of considered intervals have been processed by the accurate method. However, 7.7 % remain for an approximation. These performance indicators could be used for benchmarking and to analyze the company's evolution over time. Furthermore, some insights regarding compliance violations could encourage additional research towards the course of events within specific cases.

6 Challenge 3: Anomaly Detection

6.1 Approach

The process owner's third question deals with the detection of conspicuous and unusual process cases. On the one hand, process cases are considered anomalous if they do not fit any of the process models derived in challenge 1. On the other hand, a process case is regarded as potentially abnormal if it has either an unusual sequence of activities or extreme attributes, such as an unusually long duration, or extremely high costs. To answer the question, we applied multiple, partly interrelated approaches that address different aspects of the problem:

1. Descriptive attribute-based analysis: Based on a specific attribute, we searched the process log for conspicuous values or value combinations, i.e., process cases with unusual start and end times, or surprising long process executions.
2. Token-based replay: We used token-based replay to match the event log and the mined process models from challenge 1. All event logs that did not fit any of our mined models were considered anomalous.
3. Language model encoding: We trained a neural network to learn a language model from the case activity sequences. Next, we trained a second neural network to classify the event logs to one of the mined process models. We used the language model to encode the process cases before we passed them to the classifier. Subsequently, we compared the results of the classification task with those of token-based replay.
4. Root cause analysis: After identifying a series of abnormal process cases through 1-3, we compare the distributions of the potential anomalies and the normal process cases according to specific attributes, i.e., do the abnormal process cases often involve certain users or vendors?

6.2 Descriptive attribute-based analysis

During the detailed analysis and the filtering of the process we found some potential anomalies in the BPI log. The first revelation was that some process executions consists only of 2 events, which occurred frequently and independently of each other in terms of time (*Create Purchase Order Item* followed by *Delete Purchase Order Item*). This is particularly often done by certain users (e.g., in case 4507000392_00010, 4507000998_00040, 45070001217_00010 and several others by user 085). As these cases often take less than a minute, this may indicate an automated process or a batch processing. Another noteworthy feature is the *Create Purchase Oder Item* activity followed by *Change Price*. Those cases (e.g. 4507033253_00010 and 4507033261_ 00010) occur also in the consignment sub log and are carried out by the same users (388, 234) several times, with the start and end of the cases separated by one minute. However, the proximity of case ids could indicate a system-related error.

In the 2-way matching sub-log, we found that the most common variant of cases is the creating of a purchase order item followed by the activity *Change Approval for Purchase Order*. Additionally, those steps are always performed by user 602 and 603 and are also conspicuously close to each other or slightly offset in time (e.g. in case 4508076194_00010 or 4507075965_00040). In addition, it is noticeable that users 602, 601 and 60-65 are very often involved in conspicuously uniform processes. Here, the probability that these are not real system users is very high. When looking at the process frequencies, it was also found that there is a significant increase of the activity *Record Invoice Receipt* in the middle of the log period (e.g. case 4507075981_00010 and case 4507076007_00010). Looking at the log structures and sequences, it became clear that there are some conspicuous sequences within the log, including the continuous sequence of identical steps beyond those already described above, e.g., the activity *clear record invoice receipt* is followed by *clear invoice* and *vendor creates invoice.*

6.3 Token-based replay

By comparing the event log with the traces in the process model, token-based replay allows us to identify early deviations and have a first look at the replay ability of traces. As starting point for anomaly detection, we will use this technique to identify all the cases with potential deviations. Doing this, we focus on the measured fitness value. A fitness value of 1 means that a trace (or all traces of the event log) can be replayed without missing tokens, whereas a fitness value of 0 means that there are missing and remaining tokens. Thus, a valid execution sequence of the process model is not given.

To identify the abnormal case, we mined different process models based on six identified process models from Challenge 1 (excluding SRM). For token-based replay, we used the python package pm4py (http://pm4py.org/). The models were mined using the ProM plugin *Mine Petri net with Inductive Miner* [6], applying different noise threshold (5, 10, 20, 30). Applying token-based replay on the different petri-net models in PM4Py, we get the results shown in Figure 12.

Next, we selected only the models with the highest average trace fitness. These will be used to check conformance of each model based on the original event log. Our idea

is that we want to know which models' conformance is given and can be replayed by the event log. For those which cannot replayed, a potential anomaly could result. Figure 13 shows the results.

log	model	average_trace_fitness	log_fitness	perc_fit_traces
consignment/consignment.xes	consignment/consignment-imi-05.pnml	80.700786	0.948929	0.929467
consignment/consignment.xes	consignment/consignment-imi-10.pnml	75.982894	0.917785	0.896832
consignment/consignment.xes	consignment/consignment-imi-20.pnml	75.962202	0.910183	0.883898
consignment/consignment.xes	consignment/consignment-imi-30.pnml	88.432887	0.952468	0.935357
2-way-match/2-way match.xes	2-way-match/2-way-match-imi-05.pnml	99.137931	0.998453	0.997536
2-way-match/2-way match.xes	2-way-match/2-way-match-imi-10.pnml	99.137931	0.998453	0.997536
2-way-match/2-way match.xes	2-way-match/2-way-match-imi-20.pnml	59.961686	0.926794	0.910910
2-way-match/2-way match.xes	2-way-match/2-way-match-imi-30.pnml	50.191571	0.781656	0.658020
3-way-before/3-way-before-one-line-item.xes	3-way-before/3-way-before-one-line-imi-05.pnml	88.429760	0.947728	0.959083
3-way-before/3-way-before-one-line-item.xes	3-way-before/3-way-before-one-line-imi-10.pnml	75.041886	0.923406	0.937011
3-way-before/3-way-before-one-line-item.xes	3-way-before/3-way-before-one-line-imi-20.pnml	87.071681	0.960404	0.967427
3-way-before/3-way-before-one-line-item.xes	3-way-before/3-way-before-one-line-imi-30.pnml	99.178068	0.945271	0.960953
3-way-before/3-way-before_mult_line_item_witho...	3-way-before/3-way-before-mult-line-imi-05.pnml	75.891678	0.963476	0.965985
3-way-before/3-way-before_mult_line_item_witho...	3-way-before/3-way-before-mult-line-imi-10.pnml	48.637715	0.881452	0.885334
3-way-before/3-way-before_mult_line_item_witho...	3-way-before/3-way-before-mult-line-imi-20.pnml	0.000000	0.671858	0.645219
3-way-before/3-way-before_mult_line_item_witho...	3-way-before/3-way-before-mult-line-imi-30.pnml	0.000000	0.741095	0.713111
3-way-after/3-way-after_without_srm_service.xes	3-way-after/3-way-after_without_srm_service-im...	92.164035	0.993245	0.994919
3-way-after/3-way-after_without_srm_service.xes	3-way-after/3-way-after_without_srm_service-im...	12.244898	0.854073	0.954206
3-way-after/3-way-after_without_srm_service.xes	3-way-after/3-way-after_without_srm_service-im...	12.244898	0.852917	0.953672
3-way-after/3-way-after_without_srm_service.xes	3-way-after/3-way-after_without_srm_service-im...	2.387370	0.722460	0.812670
3-way-after/3-way-after_without_srm_without_se...	3-way-after/3-way-after_without_srm_without_se...	92.417618	0.972833	0.964196
3-way-after/3-way-after_without_srm_without_se...	3-way-after/3-way-after_without_srm_without_se...	72.176602	0.912530	0.899601
3-way-after/3-way-after_without_srm_without_se...	3-way-after/3-way-after_without_srm_without_se...	68.859977	0.895967	0.893476
3-way-after/3-way-after_without_srm_without_se...	3-way-after/3-way-after_without_srm_without_se...	68.859977	0.888001	0.890800

Figure 12: Results of Token Replay using PM4Py

	consignment-imi-30.pnml	2-way-match-imi-05.pnml	3-way-before-one-line-imi-30.pnml	3-way-before-mult-line-imi-05.pnml	3-way-after_without_srm_service-imi-5.pnml	3-way-after_without_srm_without_service-imi-5.pnml	label
0	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True
2	False	True	True	False	True	True	True
3	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True
5	True	True	True	True	True	True	True
6	True	True	True	True	True	True	True
7	True	False	False	True	False	False	True
8	True	True	True	True	True	True	True
9	False	True	False	False	True	False	True

Figure 13: Results of token-based replay of the fittest models and the original event log

A TRUE-flag means that conformance is given. If a case has only TRUE-flags, this means that no anomaly could be identified. While conformance is given for each model, the case consists of standardized activities. For cases which have a TRUE-flag as well as a FALSE-flag, deviations are identified. For us, only the cases with only FALSE-flags are interesting. These are abnormal cases because they cannot related to even one of the mined models. We could identify 250.328 compliant cases, whereas 1.406 are abnormal. These will be analyzed more in detail in the following.

6.4 Language model encoding and anomaly classification

In this section, the objective is to develop a classifier that can map the process cases to the mined process models from challenge 1 and highlights cases that are particularly difficult to assign to the process models. We argue that in some ways event logs have similar characteristics than natural language, i.e., both types have long term dependencies and hierarchical relations and we, therefore, can apply state of the art text classification techniques to business process intelligence problems. In NLP, language modeling is the task of assigning a probability to sentences in a language. Besides appointing a probability to each sequence of words, the language models also assign a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words. A language model can be developed and used standalone, such as to generate new sequences of text that appear to have come from the corpus. However, language modeling is a root problem for a large range of natural language processing tasks. More practically, language models are used on the front-end or back-end of more sophisticated models for tasks that require language understanding. The idea is, that if we interpret the activity sequences of process cases as sentences, we can apply language models on the event logs to obtain process understanding of the underlying processes. The proposed approach is highly inspired by [8] and also has some similarities with [9]. The approach consists of two steps, an unsupervised language model pretraining and a target task classifier fine-tuning. In our experiments, we used the state-of-the-art language model AWD-LSTM [10], a regular LSTM (with no attention, short-cut connections, or other sophisticated additions) with various tuned dropout hyperparameters.

For fine-tuning the classifier, we augment the pretrained language model with two additional linear blocks. Following standard practice for computer vision classifiers, each block uses batch normalization [11] and dropout, with ReLU activations for the intermediate layer and a SoftMax activation that outputs a probability distribution over target classes at the last layer. For a more detailed description of the neural network architecture, we refer the reader to [8] as the applied architecture has been reimplemented with a few minor adaptations.

We represented each process case by the sequence of its contained activities. Afterward, we sorted all cases by date and concatenated the events of all claims in a full event stream. Additionally, a unique character was inserted at the beginning of a case to represent the beginning of a process instance. The event stream was split into training and validation data by ratio 0.8. Training the language model for 5 epochs on the training data resulted in an accuracy of 0.91 on the validation data. Figure 14 visualizes the training process.

epoch	train_loss	valid_loss	accuracy	time
0	0.274873	0.264056	0.907840	02:06
1	0.248734	0.247786	0.913724	02:08
2	0.249719	0.245029	0.913685	02:09
3	0.252426	0.242458	0.913646	02:10
4	0.249909	0.243616	0.913592	02:09
5	0.270247	0.248494	0.912880	02:09

Figure 14: Training phase of the language model along with accuracy on the validation set.

By comparison, current language models usually achieve significantly lower accuracy values in the range around 0.3. This can be partly explained by a significantly lower complexity of the underlying processes in comparison to natural language. However, it also clearly shows that the language model was able to learn a deep process understanding to be able to predict the next process steps. Figure 15 show a randomly selected sample from the validation set. In the next step, we trained a classifier to map from the event stream representation of a process case to the anomaly score. In this process, we used the pretrained language model as an encoder for the classifier. To define the labels for the cases, we selected the anomalies that we identified with the descriptive analysis (6.1.) and the token-based replay (6.2). In total, we flagged 250,328 cases as normal and 1,406 cases as anomalous. Since we had significantly fewer anomalies than normal cases, we had to manually ensure that a representative number of anomalies were both available in the training set and the validation set.

First, we trained our classifier for 3 epochs while freezing the layers of the language model. Afterward, we unfroze the layers of the language model piece by piece and trained them further with a lower learning rate. This procedure allows to benefit from the general process understanding of the language model and to fine-tune the classifier subsequently to the problem of anomaly detection.

target	pred
Create Purchase Order Item Record Goods Receipt xxbos Create Purchase Order Item Vendor creates invoice Record Invoice Receipt Remove Payment	Create Purchase Order Item Vendor Goods Receipt Vendor Create Purchase Order Item Vendor creates invoice Record Goods Receipt Record Payment
Clear Invoice xxbos Create Purchase Order Item Record Goods Receipt Vendor creates invoice Record Invoice Receipt Clear Invoice xxbos Create	Clear Invoice xxbos Create Purchase Order Item Vendor Goods Receipt Vendor creates invoice Record Invoice Receipt Clear Invoice xxbos Create
Receipt Record Invoice Receipt Clear Invoice xxbos Create Purchase Order Item Vendor creates invoice Record Goods Receipt Record Invoice Receipt	Receipt Record Invoice Receipt Clear Invoice xxbos Create Purchase Order Item Vendor creates invoice Record Goods Receipt Record Invoice Receipt
Receipt Clear Invoice xxbos Create Purchase Order Item Vendor creates invoice Record Invoice Receipt Record Goods Receipt Remove Payment Block	Receipt Clear Invoice xxbos Create Purchase Order Item Vendor creates invoice Record Goods Receipt Record Goods Receipt Remove Payment Block

Figure 15: Language model based next step prediction.

epoch	train_loss	valid_loss	accuracy	time
0	0.034394	0.035732	0.994465	01:51

epoch	train_loss	valid_loss	accuracy	time
0	0.034717	0.034180	0.994478	02:08

epoch	train_loss	valid_loss	accuracy	time
0	0.029757	0.034254	0.994478	03:10

Figure 16: Training phase of the classification of anomalous process cases along with accuracy on the validation set.

After training, we reached a classification accuracy for detecting the anomalous cases of 0.995. Figure 16 prints the training process and the validation accuracy of the classification. The results show that the presented approach can distinguish between normal and anomalous cases and confirms the anomalies from sections 6.1 and 6.2.

6.5 Root cause analysis

Next to the question of identifying anomalous process cases, it is also interesting to find the causes which led to the anomalies in the first place. To reveal those root causes we performed an attribute-based comparison between all normal process cases and all potential anomalous process cases. For example, we compared the vendors that were involved in normal process cases versus the vendors that were mainly involved in potential anomalous cases. If a vendor is only involved in conspicuous process cases, this is a strong indication that he has a direct impact on the process flow. Other attributes we have examined are the case item category, the activities and the involved users. We choose those attributes, since from a business perspective, we expect them to have the highest impact.

Looking at the attribute *case Vendor*, some conspicuous vendors to which the purchase document was sent can be identified. Figure 17 shows an excerpt of the anomaly detection. Here we see that for some vendors the deviation between abnormal and normal classified cases is quite high. Having a look at the users (Figure 18), we can see that there is one big deviation and thus a hint for a real anomaly. For this user, more measures should be taken to ensure process compliance.

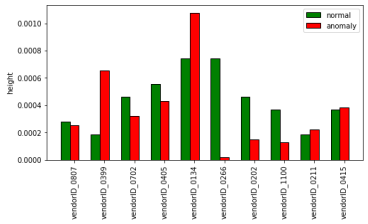

Figure 17: Excerpt results of root cause analysis for attribute "case Vendor"

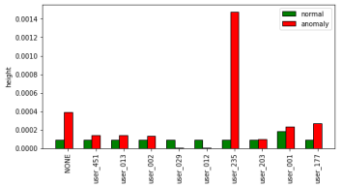

Figure 18: Excerpt results of root cause analysis for attribute "event User"

For the four different item categories, we see that no high deviations can be observed for 2-way matching and consignment. Especially for 3-way matching (invoice before GR) deviations exist, as shown in Figure 19. Doing the same for activities (Figure 20), anomalies for the two activities *Record Subsequent Invoice* and *Set Payment Block* can be observed. Compared to the other activities, the deviations are high. Thus, these activities must be reviewed.

6.6 Conclusion of the Anomaly Detection Process

In this chapter we have utilized established as well as innovative methods to identify a number of potentially anomalous process cases. For our investigations we elaborated on different methods considering case level attributes such as case item category and vendor as well as event level attributes such as time, amount and event type. In total, we were able to select 1406 potential anomalies. Subsequently, we used the selection of potential anomalies to locate different attribute deviations in the anomalies in order to uncover possible causes.

7 Conclusion

In this report, we describe our findings from the analysis of the purchase order handling process from a large coats and paints company operating from the Netherlands. We were tasked with analyzing the data in order to find conclusive answers to three leading questions regarding a collection of process models to properly represent the process, a technique to match line items, invoices and goods receipt in order to identify the throughput of a single instance, and the need for finding anomalous process behavior, i.e., instances that deviate from the prescribed process behavior. In the previous sections, we identified a collection of seven process models to describe the process. We developed a method that utilizes approximation techniques to measure the throughput time between specific activities. We combined different anomaly detection techniques to locate a selection of potential anomalous process traces and revealed different causes that led to the occurrence of the anomalies.

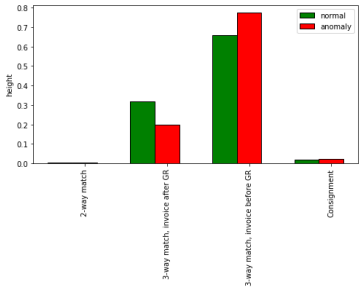

Figure 19: Excerpt results of root cause analysis for attribute "case Item Category"

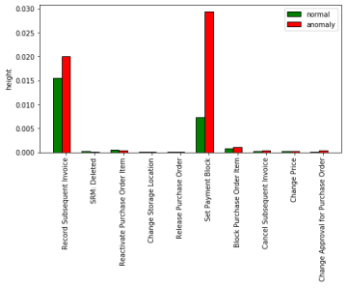

Figure 20: Excerpt results of root cause analysis for attribute "event concept:name"

Although we were able to explain some of the process behavior and context from freely-available SAP documentation, at some points we had to make assumptions. If we were to continue this analysis on a deeper level, we would prefer to talk directly to the process experts to gain an even deeper understanding of what this process entails. This would be particularly important for the second challenge, where we encountered peculiar behavior, but also for other features of the log.

Overall, we are convinced that the results from this challenge will help the case company to better understand their process, identify potential shortcomings, and optimize its purchase order handling in the future. We would like to express our gratitude to the case company for providing the data and to the BPIC committee for organizing this challenge.

References

Bose, R. P. J. C., & van der Aalst, W. (2009). Context Aware Trace Clustering: Towards Improving Process Mining Results. In *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 401–412).

Carmona, J., Dongen, B. van, Solti, A., & Weidlich, M. (2018). *Conformance Checking*

- *Relating Processes and Models*. Springer International Publishing. https://doi.org/10.1007/978-3-319-99414-7

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ArXiv Preprint ArXiv:1801.06146*.

Leemans, S., Fahland, D., & van der Aalst, W. (2014). Discovering block-structured process models from event logs containing infrequent behaviour. In N. Lohmann, M. Song, & P. Wohed (Eds.), *Business Process Management Workshops* (pp. 66–78).

Mannhardt, F., De Leoni, M., & Reijers, H. A. (2017). Heuristic mining revamped: An interactive, data-Aware, and conformance-Aware miner. *CEUR Workshop Proceedings*, *1920*.

Nolle, T., Seeliger, A., & Mühlhäuser, M. (2018). BINet: Multivariate Business Process Anomaly Detection Using Deep Learning. In *International Conference on Business Process Management* (pp. 271–287).

Thaler, T., Ternis, S., Fettke, P., & Loos, P. (2015). A Comparative Analysis of Process Instance Cluster Techniques. *Proceedings Der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015)*, 423–437. Retrieved from http://www.tom-thaler.de/publications/Thaler2015_ProcessInstanceClustering.pdf

van der Aalst, W., Adriansyah, A., & van Dongen, B. (2011). Causal nets: A modeling language tailored towards process discovery. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6901 LNCS*(1), 28–42. https://doi.org/10.1007/978-3-642-23217-6_3

van der Aalst, Wil. (2016). *Process Mining - Data Science in Action* (2nd ed.). Berlin, heidelberg: Springer Verlag. https://doi.org/10.1007/978-3-662-49851-4

Weijters, A. J. M. M., & Ribeiro, J. T. S. (2011). *Flexible heuristics miner (FHM)*. *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*. https://doi.org/10.1109/CIDM.2011.5949453

Weijters, A., van der Aalst, W., & de Medeiros, A. (2006). Process Mining with the HeuristicsMiner Algorithm. *Cirp Annals-Manufacturing Technology*, *166*(May), 1–34. https://doi.org/10.1.1.118.8288

Appendix A: Detailed table of the filtering process for the consignment sub log

Table 11: Filtering for start events of the consignment sub process

Name	Remarks
~~Change Quantity~~	Pending process with start before 2018
Create Purchase Requisition Item	
Create Purchase Order Item	

~~Record Goods Receipt~~	pending Process with start before 2018
~~Receive Order Confirmation~~	Pending Process with start before 2018, e.g. first event of 4508047120_00001 is on the 03.08.2018
~~Delete Purchase Order Item~~	Only Delete Purchase Order Item -> Create Purchase Order Item

Table 12: Filtering for end events of the consignment sub process

Name	Remarks
~~Create Purchase Order Item~~	Simultaneous start event when start in 12/2018
~~Change price~~	12 identical cases with 1 min delayed start each
~~Change Delivery Indicator~~	Many pending processes or processes at the end of 2018
~~Cancel Goods Receipt~~	Many pending processes or processes at the end of 2018
~~Change Quantity~~	Many pending processes or processes at the end of 2018
Delete Purchase Order Item	Possible end of process if something has been changed in the process (Update, Cancel, Change, Reactivate)
~~Reactivate Purchase Order Item~~	pending processes
~~Update Order Confirmation~~	pending processes
~~Receive Order Confirmation~~	pending processes, very often user 64, 30 29, 63, 65
Records Goods Receipt	
~~Change Storage Location~~	pending processes
Name	Remarks
Change Approval for Purchase Order	User_602, User_603 always executes identical steps
~~Create Purchase Order Item~~	Process at the end of 2018, end is not visible
Vendor creates invoice	
~~Vendor creates debit memo~~	Possible intermediate event with long lead time

Table 13: Filtering for start events of the 2-way-matching sub process

Name	Remarks
Change Approval for Purchase Order	User_602, User_603 always executes identical steps
~~Create Purchase Order Item~~	Pending processes at the end of 2018
Vendor creates invoice	
~~Vendor creates debit memo~~	Possible intermediate event with long lead time

Table 14: Filtering for end events of the 2-way-matching sub process

Name	Remarks
~~Change Approval for Purchase Order~~	Process at the end of 2018, end is not visible
Clear Invoice	
~~Create Purchase Order Item~~	Process at the end of 2018, end is not visible
~~Delete Purchase Order Item~~	Pending processes, often including user 602
Records Invoice Receipt	Clear accumulation in the middle of the log period
~~Set Payment Block~~	Steps which are often done by user 602 603, pending processes
~~Vendor creates invoice~~	Here follows the activity on clear invoice and record invoice receipt

Table 15: Filtering for start events of the 3-way-matching after GR sub process

Name	Remarks
Create Purchase Order Item	
Create Purchase Requisition Item	Ok but always before *Create Purchase Order Item*
Vendor creates debit memo	Part of time overlapping processes
Vendor creates invoice	

Table 16: Filtering for end events of the 3-way-matching after GR sub process

Name	Remarks
Record Service Entry Sheet	
Record Goods Receipt	Pending and not yet finished processes
Record Invoice Receipt	Pending?
~~Create Purchase Order Item~~	Pending and not yet finished processes

Clear Invoice	
~~Remove Payment Block~~	Mostly after record invoice receipt
~~Change Price~~	Pending processes
~~Change Quantity~~	Pending processes at the end of the timeframe
~~Cancel Goods Receipt~~	Pending processes
~~Cancel Invoice Receipt~~	Pending processes
~~Vendor creates debit memo~~	Pending processes
Delete Purchase Order Item	Ok, but to less cases
Change Delivery Indicator	Pending processes, but also potential end if indicator has been set to 0
Set Payment Block	Ok but to less cases
Change Approval for Purchase Order	Only after the deletion of a purchase order item
Cancel Subsequent Invoice	Only after *clear invoice*
~~Reactivate Purchase Order Item~~	To less cases
~~Change Final Invoice Indicator~~	To less cases, but possible end event

Table 17: Filtering for start events of the 3-way-matching before GR sub process

Name	Remarks
Create Purchase Order Item	
Create Purchase Requisition Item	Ok but always before *Create Purchase Order Item*
~~Vendor creates debit memo~~	Part of time overlapping processes
Vendor creates invoice	

Table 18: Filtering for end events of the 3-way-matching before GR sub process

Name	Remarks
~~Block Purchase Order Item~~	Pending process, often after *Change Approval for Purchase Order*
Cancel Goods Receipt	Ok -> Abort
Cancel Invoice Receipt	Ok -> Abort
Cancel Subsequent Invoice	Ok -> Manual Billing process
~~Change Approval for Purchase Order~~	Pending process
~~Change Currency~~	Pending process
Change Delivery Indicator	ok
~~Change Final Invoice Indicator~~	Ok, to less cases
Change Price	Ok, possible post Invoice adaption
Change Quantity	Ok, possible post Invoice adaption

~~Change Storage Location~~	Pending process
Change payment term	Pending process
Clear Invoice	
~~Create Purchase Order Item~~	Pending process
Delete Purchase Order Item	
~~Reactivate Purchase Order Item~~	Pending process
~~Receive Order Confirmation~~	Pending process
~~Record Goods Receipt~~	Pending process
Record Invoice Receipt	
~~Record Subsequent Invoice~~	Ok, but less cases
~~Release Purchase Order~~	Pending process
Remove Payment Block	
~~Update Order Confirmation~~	Pending process
Vendor creates debit memo	
Vendor creates invoice	

Appendix B: Detailed table of the filtering process for the consignment sub log

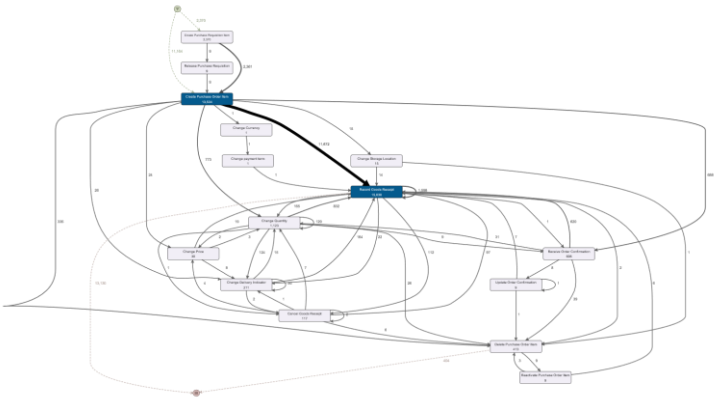

Figure 21: Process Discovery of the manually filtered consignment sub log using DISCO (100% of all activities and paths)

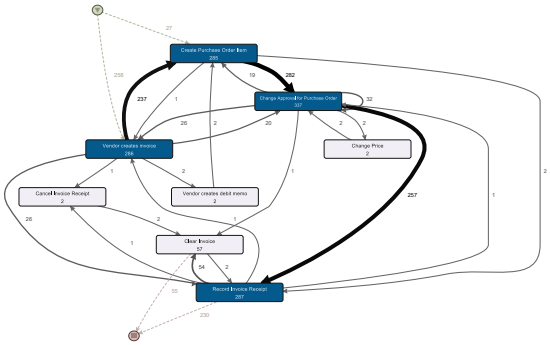

Figure 22: Process Discovery of the manually filtered 2-way-matching sub log using DISCO (100% of all activities and paths)

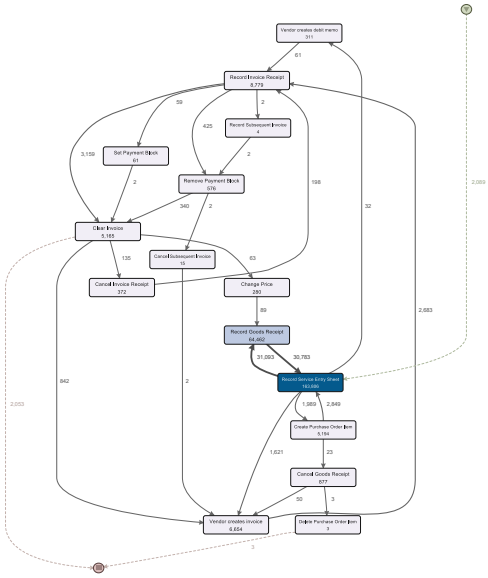

Figure 23: Process model discovered for 3-way-after service line items

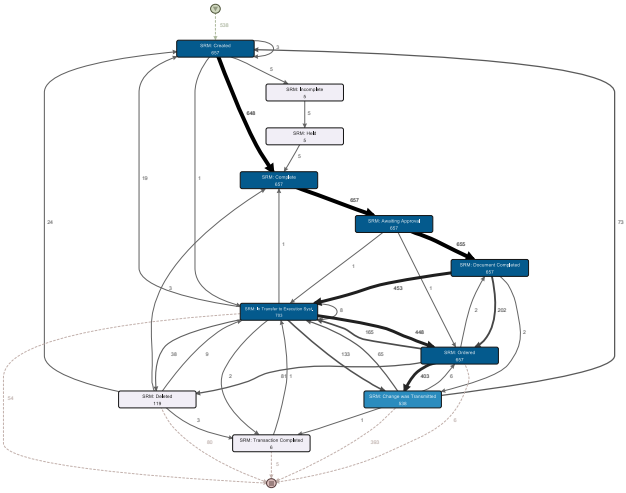

Figure 24: Process Discovery of the manually filtered 3-way-matching after GR SRM sub log using DISCO (100% of all activities and paths)

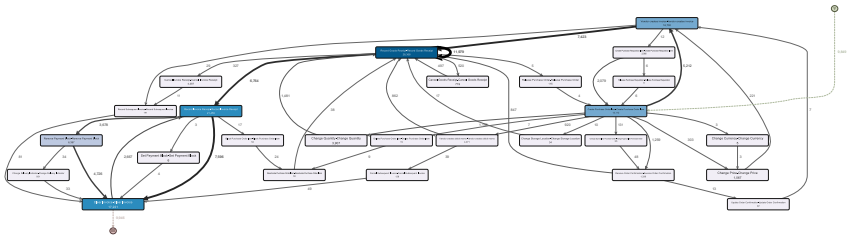

Figure 25: Discovered process model for 3-way-before-items with multiple invoices and GR

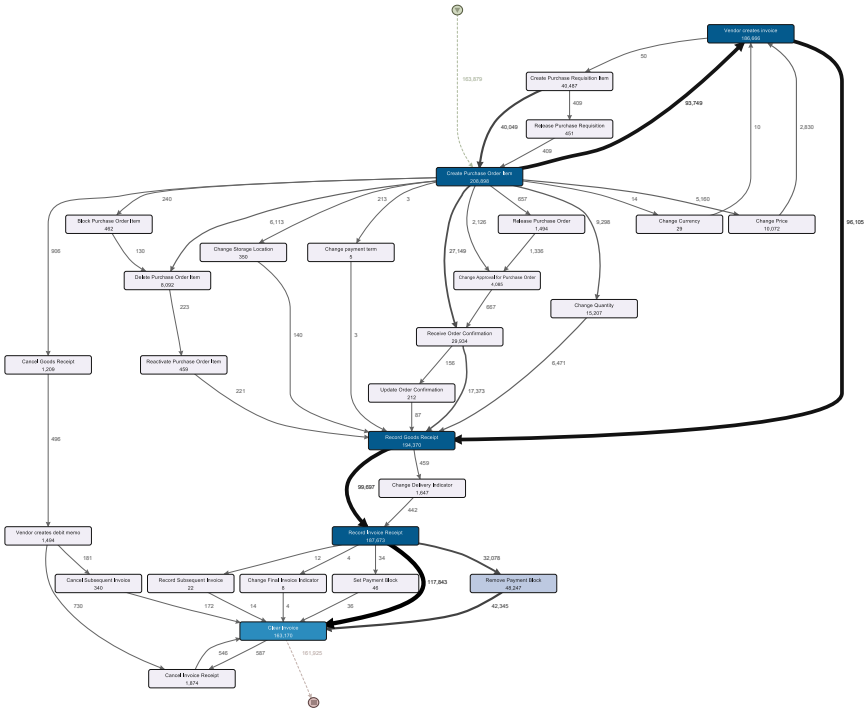

Figure 26: Discovered process model for 3-way-before items with one invoice and one GR

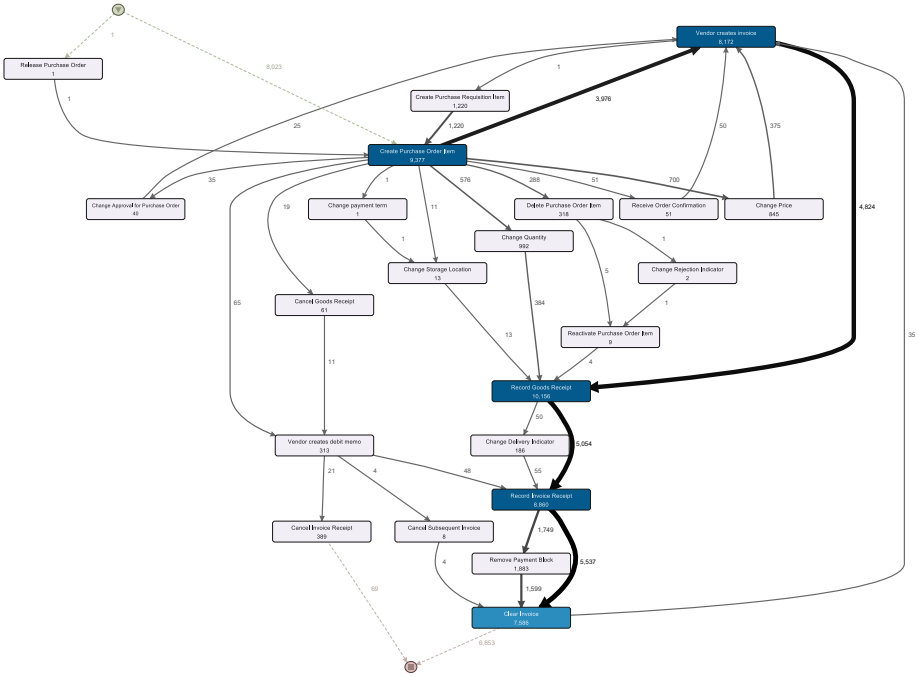

Figure 27: Process model discovered for 3-way-after non-service items