BPI Challenge 2019

Dominik Hüser, Philipp Heisenberger

RWTH Aachen University
dominik.hueser@rwth-aachen.de
philipp.heisenberger@rwth-aachen.de

Abstract. Recently, Process Mining became a major technique for auditing teams in the area of business process intelligence. Especially in purchase to pay processes (P2P process), process mining can uncover unrealized optimization potential, which again results in overall cost reduction in the company. In this year's BPI challenge we therefore deal with a P2P process log offered by a company which is operating in the context of coatings and paints.

The report deals with the detailed analysis of this log, starting with a general analysis of the unfiltered log, to gain information how to filter the log properly. The filtered log is then analyzed regarding the given case attributes. The main part will focus on the creation of a process model that describes the as-is process properly. Reasonable sublogs are defined to gain several simple and precise models rather than one large, complicated, imprecise one. We split the log according to the cases' category: SRM (supplier relationship management), 2-way matches, consignments, 3-way matches with invoice before goods receipt and 3-way matches with invoice after good receipt. The main KPIs this report focuses on are the throughput time, automation rate, rework rate as well as a social analysis. Coming from found bottlenecks and inefficiencies, we will identify deviating cases, vendors and resources to finally round up the whole report with a summary.

To mine proper models we will rely on the tools ProM, RapidProM and PM4Py. We use Celonis to gain further insights into the log, and to define proper KPIs via the Process Querying Language (PQL).

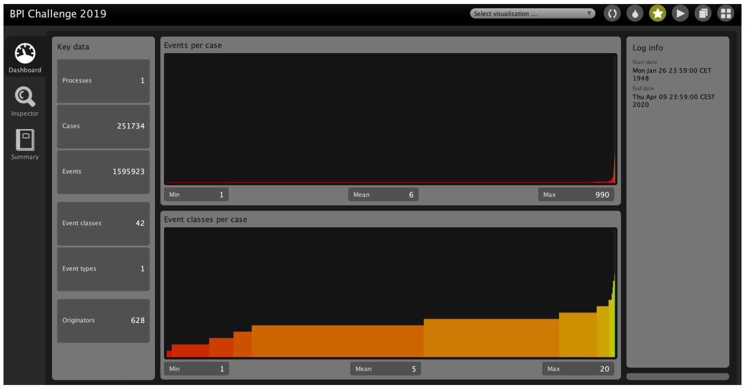
Keywords: Business Process Intelligence, Purchase to Pay, Process Mining

1 General Information

According to [2] starts a typical Purchase to Pay Process (P2P) with a purchase requisition order, which is a formal request for goods or services, followed by a vendor selection who then receives the purchase order from the company. After that, we receive the goods and receiving documents. Then the vendors sends the invoice which then is entered to the ERP. Then a three way match is performed automatically and line items don't follow that schema need to be investigated manually. After that the invoice can be cleared.

1.1 General Information about the unfiltered log

The log [3] which will be analysed in the following describes a purchase to pay process excluding the approval workflows for purchase orders and invoices. From fig. 1, obtained via Prom 6, we can find out key information about the log: We observe 251734 cases with in total 1595923 events of 42 different event classes. Every event got the same type. On average we got 6 events of 5 different event classes, variating between one and 990 events per case, while we observe between 1 and 20 different event classes per case. The log contains events which were executed between 26.01.1948 and 09.04.2020.



Key data

Processes 1

Cases 251734

Events 1595923

Event classes 42

Event types 1

Originators 628

Events per case: Min 1, Mean 6, Max 990

Event classes per case: Min 1, Mean 5, Max 20

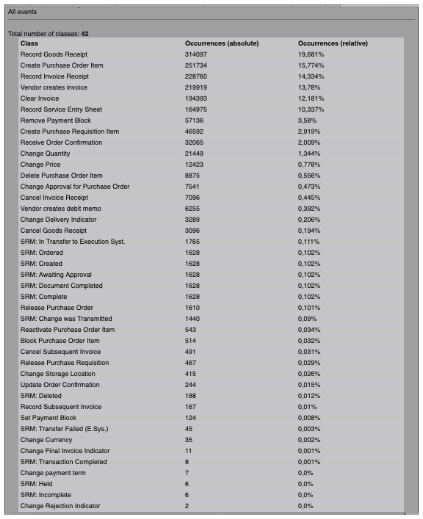
Log info

Start date
Mon Jan 26 23:59:00 CET 1948

End date
Thu Apr 09 23:59:00 CEST 2020

Fig. 1. General Information about the unfiltered log.

In fig. 2 we see the 42 different event classes and their occurrences. We observe that 32 of the the 42 have less than one percent relative occurrence. On the other hand we see that "Record Goods Receipt" (19,7%), "Create Purchase Order Item" (15,8%), "Record Invoice Receipt" (14,33%), "Vendor creates Invoice" (13,8%), "Clear Invoice" (12,2%) and "Record Service Entry Sheet" (10,3%) appear highly frequently. As we observe later those events often are executed several times per case. Additionally, we spot several event classes labeled with the prefix SRM, while activities without that prefix refer to standard purchase to pay actvities, SRM activities relates to the Supplier Relation Management.

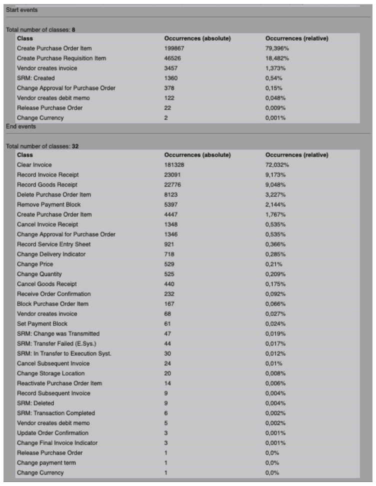


All events

Total number of classes: **42**

Class	Occurrences (absolute)	Occurrences (relative)
Record Goods Receipt	314097	19,681%
Create Purchase Order Item	251734	15,774%
Record Invoice Receipt	228760	14,334%
Vendor creates invoice	219919	13,78%
Clear Invoice	194393	12,181%
Record Service Entry Sheet	164975	10,337%
Remove Payment Block	57136	3,58%
Create Purchase Requisition Item	46592	2,919%
Receive Order Confirmation	32065	2,009%
Change Quantity	21449	1,344%
Change Price	12423	0,778%
Delete Purchase Order Item	8875	0,556%
Change Approval for Purchase Order	7541	0,473%
Cancel Invoice Receipt	7096	0,445%
Vendor creates debit memo	6255	0,392%
Change Delivery Indicator	3289	0,206%
Cancel Goods Receipt	3096	0,194%
SRM: In Transfer to Execution Syst.	1765	0,111%
SRM: Ordered	1628	0,102%
SRM: Created	1628	0,102%
SRM: Awaiting Approval	1628	0,102%
SRM: Document Completed	1628	0,102%
SRM: Complete	1628	0,102%
Release Purchase Order	1610	0,101%
SRM: Change was Transmitted	1440	0,09%
Reactivate Purchase Order Item	543	0,034%
Block Purchase Order Item	514	0,032%
Cancel Subsequent Invoice	491	0,031%
Release Purchase Requisition	467	0,029%
Change Storage Location	415	0,026%
Update Order Confirmation	244	0,015%
SRM: Deleted	188	0,012%
Record Subsequent Invoice	167	0,01%
Set Payment Block	124	0,008%
SRM: Transfer Failed (E.Sys.)	45	0,003%
Change Currency	35	0,002%
Change Final Invoice Indicator	11	0,001%
SRM: Transaction Completed	8	0,001%
Change payment term	7	0,0%
SRM: Held	6	0,0%
SRM: Incomplete	6	0,0%
Change Rejection Indicator	2	0,0%

Fig. 2. General information about the unfiltered log.



Start events

Total number of classes: **8**

Class	Occurrences (absolute)	Occurrences (relative)
Create Purchase Order Item	199867	79,396%
Create Purchase Requisition Item	46526	18,482%
Vendor creates invoice	3457	1,373%
SRM: Created	1360	0,54%
Change Approval for Purchase Order	378	0,15%
Vendor creates debit memo	122	0,048%
Release Purchase Order	22	0,009%
Change Currency	2	0,001%

End events

Total number of classes: **32**

Class	Occurrences (absolute)	Occurrences (relative)
Clear Invoice	181328	72,032%
Record Invoice Receipt	23091	9,173%
Record Goods Receipt	22776	9,048%
Delete Purchase Order Item	8123	3,227%
Remove Payment Block	5397	2,144%
Create Purchase Order Item	4447	1,767%
Cancel Invoice Receipt	1348	0,535%
Change Approval for Purchase Order	1346	0,535%
Record Service Entry Sheet	921	0,366%
Change Delivery Indicator	718	0,285%
Change Price	529	0,21%
Change Quantity	525	0,209%
Cancel Goods Receipt	440	0,175%
Receive Order Confirmation	232	0,092%
Block Purchase Order Item	167	0,066%
Vendor creates invoice	68	0,027%
Set Payment Block	61	0,024%
SRM: Change was Transmitted	47	0,019%
SRM: Transfer Failed (E.Sys.)	44	0,017%
SRM: In Transfer to Execution Syst.	30	0,012%
Cancel Subsequent Invoice	24	0,01%
Change Storage Location	20	0,008%
Reactivate Purchase Order Item	14	0,006%
Record Subsequent Invoice	9	0,004%
SRM: Deleted	9	0,004%
SRM: Transaction Completed	6	0,002%
Vendor creates debit memo	5	0,002%
Update Order Confirmation	3	0,001%
Change Final Invoice Indicator	3	0,001%
Release Purchase Order	1	0,0%
Change payment term	1	0,0%
Change Currency	1	0,0%

Fig. 3. General information about the unfiltered log.

In fig. 3, we see that nearly 80% of all cases start with the creation of an Purchase Order Item, while 18% start with the creation of a Purchase Requisition Item. Much less frequently a process starts with a creation of the invoice by the vendor (1,4%). 72% of all cases then end with ”Clear Invoice”, 9,2% with ”Record Invoice Receipt” and 9% with ”Record Goods Receipt”. Much less often a case ends with other events. We need to keep in mind, that this is a real log containing information about year 2018. Cases which were not completed in the logged timeframe are now falsely appearing as complete cases with an end activity from the middle of the actual incomplete case. How we deal with that issue will be described later on.

The process is highly complex with in total 11973 different case variants. The four most common variant can be seen in fig. 4.

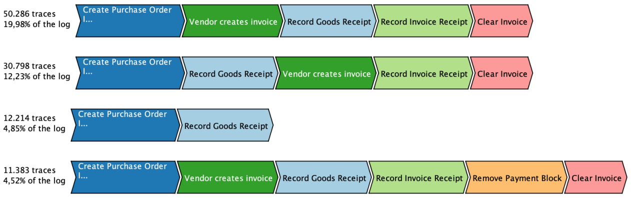


Fig. 4. Most common variants in the unfiltered log.

1.2 Filter the log

In the previous section we already talked about the time frame. We observe that there are cases reaching back to 1946 or already contain event executions from the future, namely from the year 2020. In a first step we filter the cases on the timestamps of the first event which must be executed on the 01.01.18 or later. In a second step we filter cases such that we only observe events where the last event is executed at latest in present, so that we won't regard cases containing future events. We filtered via a prom4py python script.

As already mentioned, do we also need to deal with filtering incomplete cases. This task is not as easy as it seems since incomplete cases can easily be confused with infrequent behavior of the process. We look into fig. 3 again to get information about the frequency of end activities. We immediately spot candidate endings which might have resulted from incompleteness (cf. ”Release Purchase Order”, ”Change Payment Term”, ”Change Currency”, each occurring only once in the end of a trace). We still decided to keep those traces inside the log and deal with them later, by leaving out a certain part of the less frequent cases when mining. The advantage of doing the filtering on the fly while mining is the possibility to play with the severeness of the filter operation and see the impact on the model much better, so that we hopefully find a much better fitting model describing the process more precisely. After filtering we lost 265 cases. That reduces the maximum number of events per case from 990 to 868 events per case. And the maximum amount of event classes per case from 20 to 19. Additionally, we eliminated 61 case variants.

1.3 Insight into the log

After filtering the log for outliers we can now make some basic observations about the given cases. We primarily use Celonis, Prom 6 and pm4py to perform that task.

Let us first regard the PO Document Types (cf. fig. 5): We see that by far most PO documents are standard PO documents (74598 docs), 978 are EC Purchase Orders and 704 are Framework Orders. If we look at the PO Item types (fig. 5), we see that 87% of all cases are Standard PO Items, while 6% of all cases are Consignment and 2% are Service PO Items.

Company When we regard the subsidiary from where this purchase originated, which is defined in the `Case-Company` attribute of an item, we spot the following distribution in fig. 6. Nearly all items are related to companyID_0000. Only 1027 (a share of 0,41%) are related to companyID_0003. CompanyID_0001 and companyID_0002 are only related to two items each. We want to dig deeper into those three less common cases, quickly: In fig. 7 we see dotted charts regarding the company and several other criteria. What we basically conclude is the following: For some reasons we don't have any information about the document type, GR Indicator, Item Category nor the spend area for comapnyID_0001 and

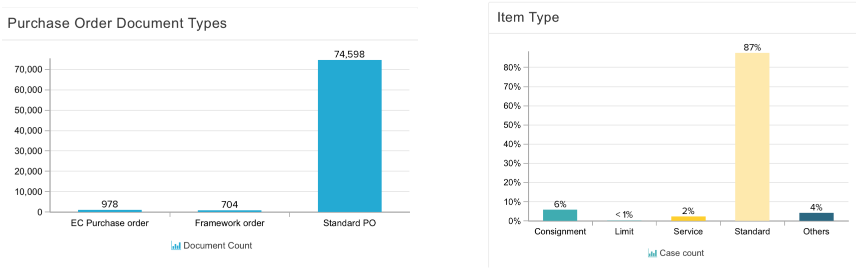


Fig. 5. Spend Area.

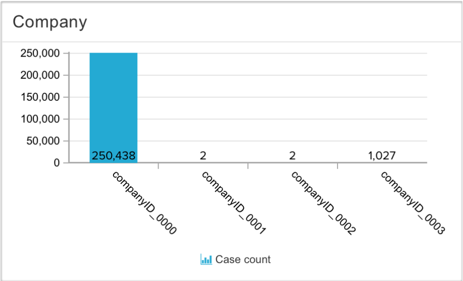


Fig. 6. Most common variants in the filtered log.

comapnyID_0002. The reason for that might be in the process: In all of those four cases the PO Item was deleted in the end of the process.

When we compare companyID_0000 (in the following called company 0) and companyID_0003 (in the following called company 3) then we observe that company 3 is related to Framework Orders only, while company 0 is related to all three types. Related to company 3 are only PO Items without GR. This also leads to the obvious observation that company 3 only deals with 2-way match items, as visible in fig. 7. This observation again fits to another observation we made earlier: Only Framework Orders are related to company 3.

When we compare the spend class of company 0 and 3 we see that company 3 only performs purchase orders of class NPR and OTHER while company 0 contains all kind of PO Item classes: NPR, PR, OTHER and NULL. When we dig deeper here we see that company 3 exclusively performs PO Items of spend area "Real Estate" and "Energy". Both companies perform items of spend area "Enterprise Services", "CAPEX & SOCS" , "Workforce Services" and "Others". Those are basic purchases which need to be made in every company. Excluding those areas, we can split company 0 and 3 in following two clusters:

- **Company 0:** Actual production (Titanium Dioxide, Latex and Monomers, Pigments and Colorants, Solvents, Specialty Resins, Commodity Resins, Additives) and shipping and related (Logistics, Packing, Sales, Marketing)
- **Company 3:** Real Estate, Energy

Let us in the following look closer into the spend areas of the company.

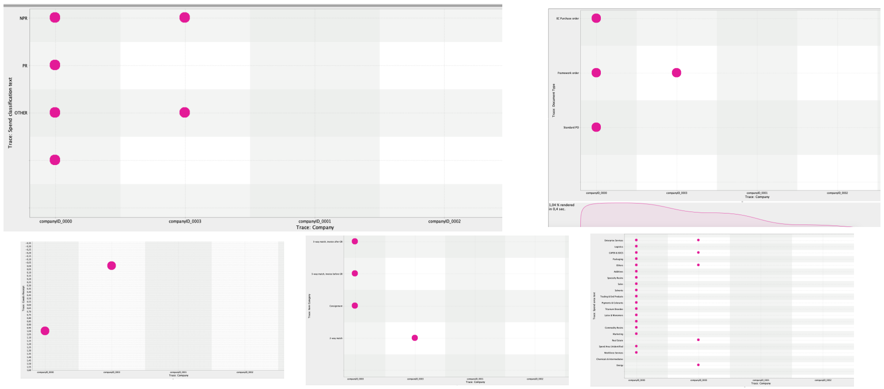


Fig. 7. Several comparisons betweent the company and a) the class, b) the doctype, c) the GR, d) the Item Spend category and e) the Spend Category Text

Spend Areas In fig. 8 we see that most cases are assigned to class "PR" (more than 160000), followed by "NPR" (more than 80000) and then followed by "OTHERS" and NULL valued. If we regard the Spend Area explicitly then we see than most cases deal with "Packing" (109181), Sales (65790), Trading and End Products (22204) followed by "Additives" and others.

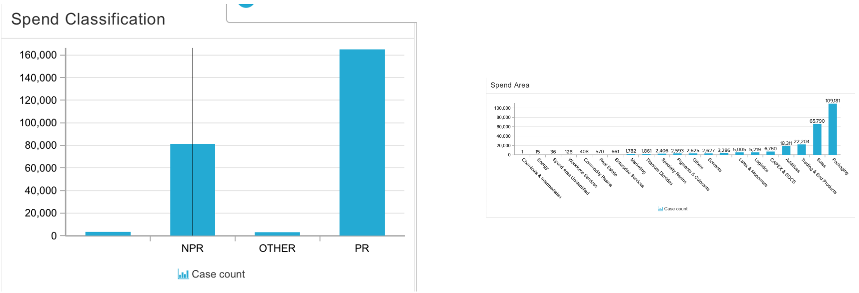


Fig. 8. Spend Classification and Area.

Vendors When we regard the vendors we spot in total 1965 vendors. In fig. 9 we see how many vendors each area of spend got in the log. We see that especially many different vendors are present in the area of "Sales", "Additives", "Packing" and "Marketing". Interesting is the amount of one time vendors: In total we saw that there are 432 vendors where we performed a purchase order only once in the given time frame.

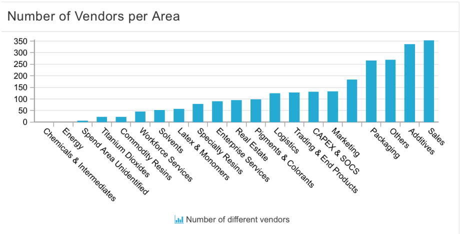


Fig. 9. Number of different vendors per spend area.

Matching When we look at the case count of each of the categories we see that by far (220811 times) most items are 3 way matches where the invoice was received before the good. Followed by 3-way matches where the good arrived before the invoice (15134 times). Tightly again followed by Consignments (14497 occurrences) and 2-way matches (1027 times). Interesting is the distribution of the actual total value of the purchase order items. We therefore summed up the net value of the creation of a purchase order item. Here we see that most value is actually in the category 3-way match with invoice after GR, followed by 3-way matches where the invoice arrived before the goods. In the previous analysis we saw that all 2-way matches were performed by company 3 and that those are all framework orders. That makes sense since it primarily deals with purchases of non-material goods, and therefore no goods are receipt. In the dotted chart below we additionally see that some framework orders are actually processed as 3 way match where the invoice arrived after the GR. Standard POs are either 3-matches or Consignments while EC Purchase Orders are always 3-way-matches.

1.4 Usability of the log

Due to the large variety of cases inside the log, we prefer to split the log into sublogs and then create one model for each sublog, to gain more detailed insight and simpler models. How we split further is described in the next section.

2 Model based Mining

In fig. 11 we see the distribution of throughput times. On average the throughput time is 70 days. We see additionally that there are cases, that take 383 days. To dig deeper now we derive sublogs and submodels from the overall log.

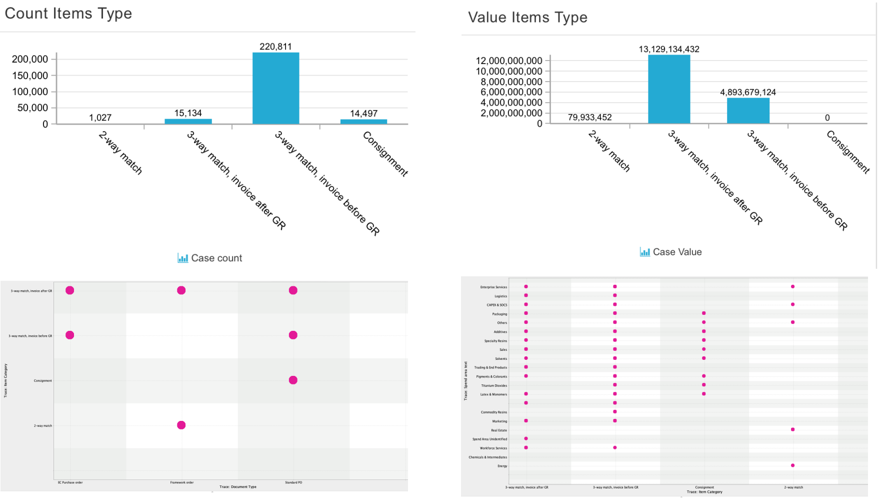


Fig. 10. Item Type Analysis.

2.1 Split data into sublogs and general mining procedure

The way we split the filtered log is visible in fig. 12. Since we first want to focus on the pure P2P process we filter out cases including SRM events. Those can then be analyzed separately, later on. The SRM filtered log is then split via the case attribute `Case-category`. The three categories are "3-way match (invoice before GR)", "3-way-match (Invoice after GR)", "Consignment" and "2-way-match".

In the following we will mine one model for each of those subsets in Prom:

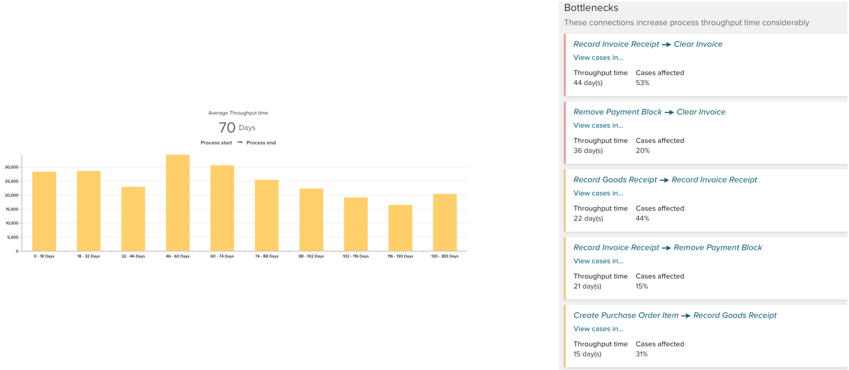
1. Mine a basic model with the inductive miner on ca. 90% of the most common traces.
2. Regard the traces and derivations and manually adapt the model if necessary.
3. Calculate the fitness and precision via the Multi Perspective Process Explorer / RapidProm and adapt the model manually if necessary.

That resulted in the following models:

2.2 Model A: 2-way match

Here we observe those traces which are marked as two way matches. In that case IV = false and GR = false. According to [1] we expect to receive only an invoice and no goods.

The Model In fig. 13 we see the mined model: Optionally the vendor starts with the creation of a debit memo, which happens rather infrequent as we see later. After that either 1) concurrently multiple changes to the approval for the



Bottlenecks

These connections increase process throughput time considerably

Record Invoice Receipt → Clear Invoice
View cases in...
Throughput time 44 day(s)
Cases affected 53%

Remove Payment Block → Clear Invoice
View cases in...
Throughput time 36 day(s)
Cases affected 20%

Record Goods Receipt → Record Invoice Receipt
View cases in...
Throughput time 22 day(s)
Cases affected 44%

Record Invoice Receipt → Remove Payment Block
View cases in...
Throughput time 21 day(s)
Cases affected 15%

Create Purchase Order Item → Record Goods Receipt
View cases in...
Throughput time 15 day(s)
Cases affected 31%

Fig. 11. General Throughput Time Distribution (left) and Bottlenecks by Celonis (right)

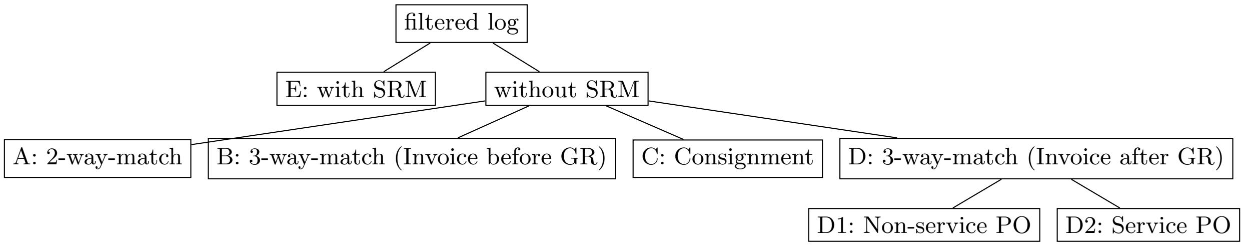
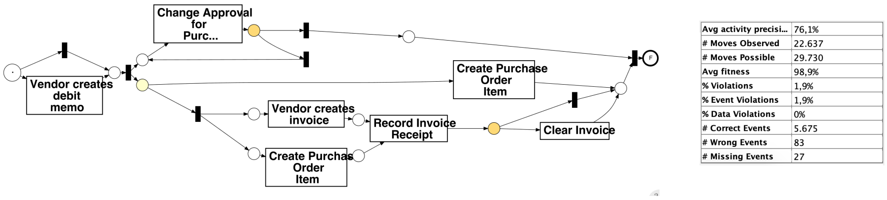


Fig. 12. Splitting the log into sublogs.

purchase order item and one creation of a purchase order item happens. Or 2) concurrently multiple changes to the approval for the purchase order item, a creation of the invoice by the vendor and the sequential flow ”Create Purchase Order Item”, ”Record Invoice Receipt” and optionally a ”Clear Invoice” happens. As expected we don’t observe any Goods Receipt events as we deal with all 2-way matches, only.

Conformance On first sight we can easily see that this model is fairly simple and easy to understand. In fig. 13, we can additionally spot the fitness as well as the precision of the model on the sublog. We got a very good fitness of 98,9%. On the other hand we observe a precision of ca. 76,1%. In fig. 13 we can immediately spot the places where we lack in precision. Main reason for the lack of precision is the loop at ”Change Approval for Purchase Order Item” since it allows a lot more behavior than actually visible in the log. Same holds for the place after the first concurrency.

Throughput and Flow In fig. 14 (up) we see the traces as well as the derivations from the previously mined model for two way matches.


Change Approval for Purc...
Vendor creates debit memo
Create Purchase Order Item
Vendor creates invoice
Record Invoice Receipt
Clear Invoice
Create Purchase Order Item

Avg activity precisi...	76,1%
# Moves Observed	22.637
# Moves Possible	29.730
Avg fitness	98,9%
% Violations	1,9%
% Event Violations	1,9%
% Data Violations	0%
# Correct Events	5.675
# Wrong Events	83
# Missing Events	27

Fig. 13. Model for 2-way matches.

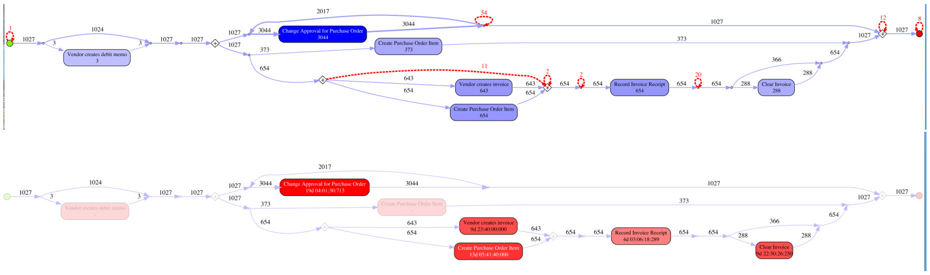

Vendor creates debit memo 3
Change Approval for Purchase Order 3044
Create Purchase Order Item 373
Vendor creates invoice 643
Create Purchase Order Item 654
Record Invoice Receipt 654
Clear Invoice 288
1027
1024
3
2017
3044
373
654
643
366
288
54
11
2
20
12
8
1
Vendor creates debit memo -
Change Approval for Purchase Order 19d 04:01:30:713
Create Purchase Order Item -
Vendor creates invoice 9d 23:40:00:000
Create Purchase Order Item 13d 05:41:40:000
Record Invoice Receipt 4d 03:06:18:289
Clear Invoice 9d 22:50:26:250

Fig. 14. Case Count and Throughput of Model A.

We see that only in 3 cases a debit memo was created by the vendor. 373 times a purchase order item was created while neither an invoice was created, received nor cleared. The other 654 cases invoices were handled. In 288 of those cases the invoice was finally cleared, while in the other 366 cases it was not. In total 3044 changes to the approval of the PO were performed, often multiple times per case. When we regard the deviations we spot no especially high number.

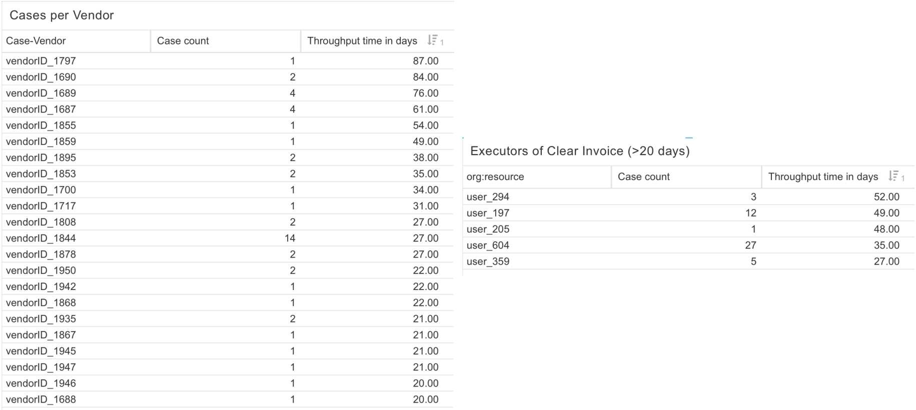
When we regard the throughput time of the other hand we directly see that the time between "Record Invoice Receipt" and "Clear Invoice" is 9 days and 22 hours on average.

When we filter on those cases, which got a significant higher throughput time here we spot the following: In fig. 15 we see the vendors and their average throughput time between recording of the invoice receipt and the payment. Focused on those vendors where payment took especially long time. Additionally we see those resources which are mostly clearing the invoices where the throughput is large.

We also see large waiting times for "Create Purchase Order Item", and "Change Approval for PO" of 13 and 19 days, which can't really be interpreted in this graphic though, due to the concurrent nature of the two activities.

2.3 Model B: 3-way match, invoice before goods receipt

Here we observe those traces which are marked as 3-way match, invoice before goods receipt. In that case IV = false and GR = true. From the BPI challenge project site [1] we take the following information: We expect to observe PO Items that require a GR while they don't require invoicing based on the GR. If



Cases per Vendor

Case-Vendor	Case count	Throughput time in days
vendorID_1797	1	87.00
vendorID_1690	2	84.00
vendorID_1689	4	76.00
vendorID_1687	4	61.00
vendorID_1855	1	54.00
vendorID_1859	1	49.00
vendorID_1895	2	38.00
vendorID_1853	2	35.00
vendorID_1700	1	34.00
vendorID_1717	1	31.00
vendorID_1808	2	27.00
vendorID_1844	14	27.00
vendorID_1878	2	27.00
vendorID_1950	2	22.00
vendorID_1942	1	22.00
vendorID_1868	1	22.00
vendorID_1935	2	21.00
vendorID_1867	1	21.00
vendorID_1945	1	21.00
vendorID_1947	1	21.00
vendorID_1946	1	20.00
vendorID_1688	1	20.00

Executors of Clear Invoice (>20 days)

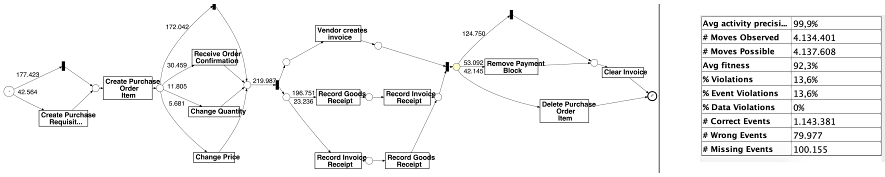
org:resource	Case count	Throughput time in days
user_294	3	52.00
user_197	12	49.00
user_205	1	48.00
user_604	27	35.00
user_359	5	27.00

Fig. 15. Throughput (≥ 20 days) from "Record Invoice Receipt" to "Clear Invoice" in Model A

we record goods receipt then its ok that this happens after the invoice receipt. But the invoice should be blocked until the goods arrived (unblock). Invoices should only be cleared if goods are received and the value matches with the invoice and the value at creation of the item.

The Model In fig. 16 we see the mined model. The process starts with an optional creation of a purchase requisition item (19% of the cases). That is followed by the creation of the PO item and an xor split: Either we change the quantity or the price, receive order confirmation or do none of those. After that, in concurrency the vendor creates the invoice and we receive goods and the invoice. In 90% of the cases the goods receipt is reported before the invoice receipt. In 10% it is the other way around. In the end either the invoice is cleared direct or after a removal of a payment block or the PO item is deleted.

Conformance As already observed, is the model fairly simple. In fig. 16 we also see the fitness and precision of the model. With our model we gain a fitness of 92,3% and a precision of 99,9%.



Avg activity precisi...	99,9%
# Moves Observed	4.134.401
# Moves Possible	4.137.608
Avg fitness	92,3%
% Violations	13,6%
% Event Violations	13,6%
% Data Violations	0%
# Correct Events	1.143.381
# Wrong Events	79.977
# Missing Events	100.155

Fig. 16. Model for 3-way match, invoice before goods receipt.

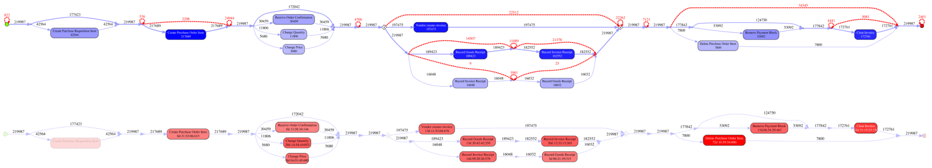


Fig. 17. Case Count and Throughput of Model B.

Frequencies We see that 80% of the cases do not contain a purchase order requisition item. We additionally see that not all cases in this log contains an order confirmation. We will look deeper into this insight later since it could result from the not perfectly fitting nature of the model. Most interesting is the later xor split: We see that 92,2% of the cases contain record goods receipt before record invoice receipt, while the other cases contain the two events ordered the other way around. In 7800 cases the purchase order item is deleted in the end. In 53092 cases a payment block needs to be removed before the invoice is cleaned finally.

Throughput Times When we look at the relevant throughput times now we see the following: When goods are receipt first, then it takes on average 20 days until the invoice is received. When the invoice is received first, then it only takes 3 days on average until the goods are received. When the purchase order item is deleted, it takes additional 72 days on average. This is definitely a bottleneck. In fig. 18 we see the throughput time from Record Goods/Invoice Receipt. In case when invoices are received before the goods, then on average 59 days pass until the invoice is cleared. In the other case that the goods are received before the invoice another 84 days pass. In general we can therefore says that the second case is significantly quicker in terms of throughput time. One reason for the high throughput times might be the mapping of items inside a purchase order. In this report, we calculate the throughput time with respect to the first occurrence of the recording and the last occurrence of the invoice clearing. This definitely needs to be kept in mind when regarding those times. We additionally want to check when invoices are cleared and when especially not, even though goods and invoices were received. This is done, now:

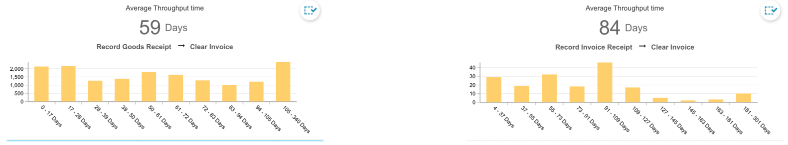


Fig. 18. Throughput of Model B from Good Receipt to Clear invoice when goods were received after the invoice (left) and from Invoice Receipt to Clear Invoice Receipt, when invoice was received after goods.

Derivations We additionally see the derivations: Those mostly result from traces where only a PO Item was created after an optional PO Requisition Item and then the case was over or goods were received but no invoice. The result are derivations at the first concurrent block where invoices and goods are received and in the end when usually a PO Item is deleted or payed. We therefore look again into the log, to spot cases where not both "Record Goods Receipt" and "Record Invoice Receipt" appeared. There are ca. 14000 cases from 747 vendors following that pattern. Those are especially interesting because the company thinks goods and invoices are both checked against each other in this sublog. Mostly Standard PO from company 0 with standard item type follow that pattern. In fig. 19 we additionally see that those cases usually are out of the PR and NPR class and additionally those vendors which are most commonly executing such cases. Interesting are those cases which additionally contain an invoice payment, even though not both record events occur in the log. We see that especially user 002 performs such payments the reason here is that user 002 performes most invoice clearances.

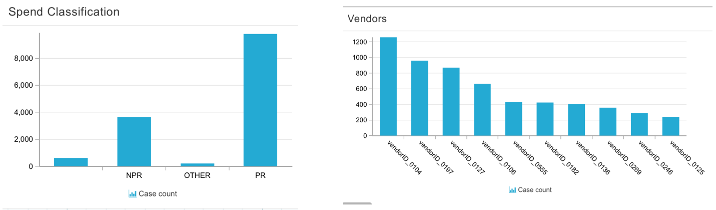
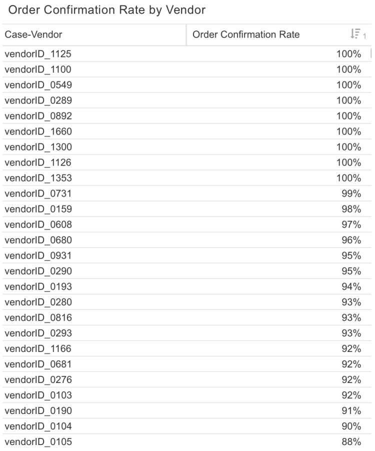


Fig. 19. Cases where not both, invoice and goods were received.

Order Confirmation Rate In fig. 21 we see the order confirmation rate of vendors (sorted by confirmation rate) of this sublog. The order confirmation rate is the fraction of cases containing an order confirmation. We defined a KPI in Celonis for the Order Confirmation Rate (cf. fig. 20) to generate that insight. Out of the 1374 vendors relevant for this sublog only 43 can hold an order confirmation rate above 70%. When we look at cases where the invoice came before the goods, then only vendor 599 (100%), vendor 120 (23%) vendor 136 (3%) perform order confirmations.

```
AVG(CASE WHEN MATCH_ACTIVITIES(NODE['Receive Order Confirmation'] ) > 0
THEN 1.0 ELSE 0.0 END)
```

Fig. 20. PQL for Order Confirmation Rate.



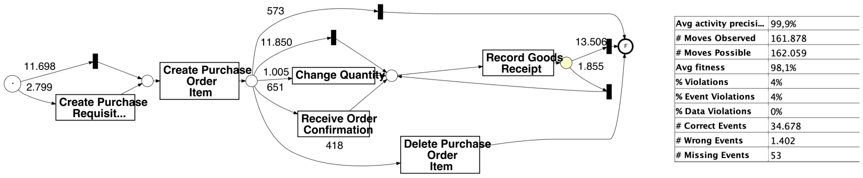
Order Confirmation Rate by Vendor

Case-Vendor	Order Confirmation Rate
vendorID_1125	100%
vendorID_1100	100%
vendorID_0549	100%
vendorID_0289	100%
vendorID_0892	100%
vendorID_1660	100%
vendorID_1300	100%
vendorID_1126	100%
vendorID_1353	100%
vendorID_0731	99%
vendorID_0159	98%
vendorID_0608	97%
vendorID_0680	96%
vendorID_0931	95%
vendorID_0290	95%
vendorID_0193	94%
vendorID_0280	93%
vendorID_0816	93%
vendorID_0293	93%
vendorID_1166	92%
vendorID_0681	92%
vendorID_0276	92%
vendorID_0103	92%
vendorID_0190	91%
vendorID_0104	90%
vendorID_0105	88%

Fig. 21. Order Confirmation rate in sublog B. (PQL in fig. 20).

2.4 Model C: Consignment

In the consignment subprocess we expect that there is no invoice handling present, since this is performed by another system [1]. GR is true and IV is false. This is indeed the case as we see in the following mined model:


573
11.850
11.698
2.799
Create Purchase Requisit...
Create Purchase Order Item
1.005
651
Change Quantity
Receive Order Confirmation
418
Delete Purchase Order Item
Record Goods Receipt
13.506
1.855
F

Avg activity precisi...	99,9%
# Moves Observed	161.878
# Moves Possible	162.059
Avg fitness	98,1%
% Violations	4%
% Event Violations	4%
% Data Violations	0%
# Correct Events	34.678
# Wrong Events	1.402
# Missing Events	53

Fig. 22. Model for consignment.

The Model At the beginning of the process we again spot an optional "Purchase Order Requisition Item Creation". 19% of the cases contain that event. The other 81% start with "Create Purchase Order Item" directly. After that 4% of the cases stop immediately without any execution in the model. 2,88% then finish with "Delete Purchase Oder Item". For 4,5% of our cases we receive an order confirmation, for 6,9% a Quantity Change is necessary the other 81,7 % go directly to the next step. After that "Record Goods Receipt" is executed in 1855 cases this happens several times per case.

To be sure that there is no trace which contains invoice handling events, which is not conforming with the mined model, we plot in fig. 23 the occurring actvities and their frequency. As expected, we observe that especially Create PO

Item and Record Goods Receipt are executed in nearly every case. Additionally, we see that there is no execution of an invoice related activity.

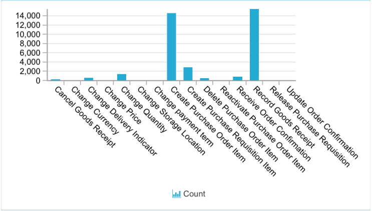


Fig. 23. Activity Count of Consignment Sublog

Conformance The mined model has fitness of 98,1% and precision of 99,9%. Therefore, represents the given subprocess quite accurately. The model is additionally simple which is a good feature for the further analysis. We see that 53 events are missing. Those can be tracked in fig. 23 again.

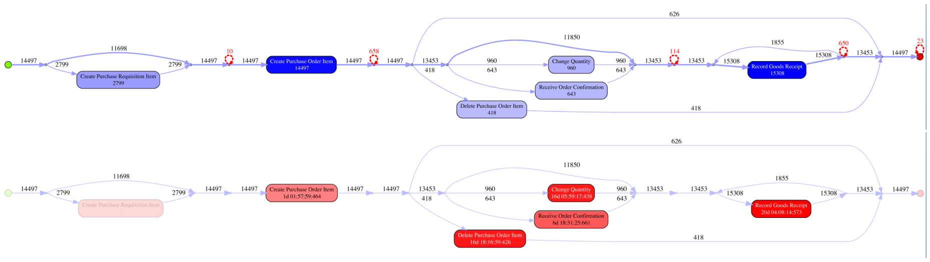

Create Purchase Requisition Item
2799
Create Purchase Order Item
14497
Change Quantity
960
Receive Order Confirmation
643
Delete Purchase Order Item
418
Record Goods Receipt
15308
10
658
114
650
23
626
11850
11698
1855
Create Purchase Requisition Item
Create Purchase Order Item
1d 01:57:59:464
Change Quantity
16d 05:59:17:438
Receive Order Confirmation
6d 18:31:25:661
Delete Purchase Order Item
16d 18:16:59:426
Record Goods Receipt
20d 04:08:14:573

Fig. 24. Case Count and Throughput of Model C.

Flow and Derivations In fig. 24 we see the deviations and throughput time of our model w.r.t. the log. We see deviations after "Create Purchase Order Item" and "Record Goods Receipt". After Creation of PO Item we see 650 derivation, those are "Change Delivery Indicator" (229) and "Record Goods Receipt" (144) which are executed additionally to the models service. After "Received Goods" again around 650 derivations are visible. Mostly "Change Delivery Indicator" (276) and "Cancel Goods Receipt" (207) executions.

Throughput When we created a Purchase Requisition Item it takes on average one day until the corresponding Purchase Order Item was created. When the PO Item is then deleted, it takes more than 16 days on average to perform that task. Same holds for changes to the quantity: 16 days pass till this is executed. Order Confirmations are received after on average 7 days. Before ”Record Goods Receipt” is executed, on average 20 days pass. We want to dig deeper into this in fig. 25, where we look into those case with especially large throughput time. When we look into the spend areas of extraordinary large throughput cases then especially ”Packing” holds a large share here. Reason for that is the generally high occurrence of ”Packing” cases in the log (cf. fig. 8). In the second largest spend area Titanium Dioxides, one third of those traces in the log end up as Consignment Cases with large throughput. Additionally, we spot those vendors which are most commonly occurring in high throughput time cases.

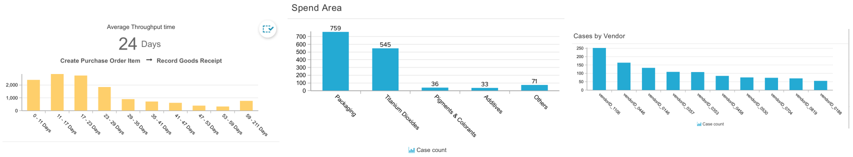


Fig. 25. Throughput between Create PO Item (first execution) and Record Goods Receipt (last execution), and the spend areas and vendors with especially high throughput (40 days or more)

Order Confirmation and other Insights Other interesting information can be found in fig. 26. In the first sub-figure we see those resources performing creations of PO items which later are deleted. In the second figure we see those vendors which most often perform order confirmation (in %). We see that there vendors which always confirm orders in the consignment sublog. Last but not least we want to look into those cases which stop after the execution of Create PO Item. Vendors which are most involved in those PO items can be found in the third graph of fig. 26.

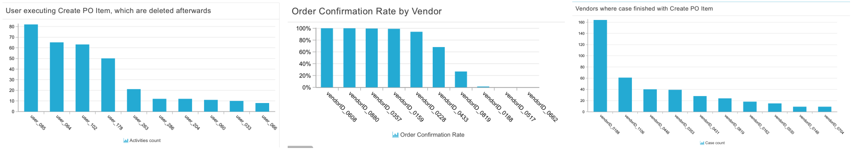
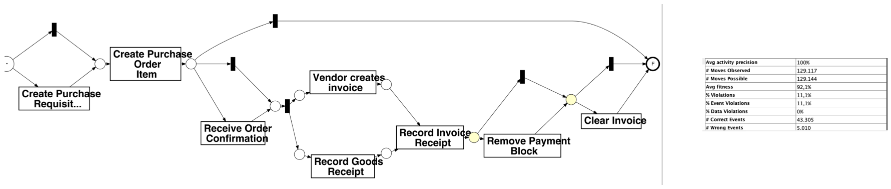


Fig. 26. a) user executing Create PO Items which are later deleted, b) Order Confirmation Rate by Vendor, c) vendors where a case finishes with PO Item Creation

2.5 Model D: 3-way matching, invoice after goods receipt

For those items we report goods receipt first and then an invoice receipt [1]. Invoices are only cleared if the values matched. IV = true and GR = true. Let us have a look at the mined model.

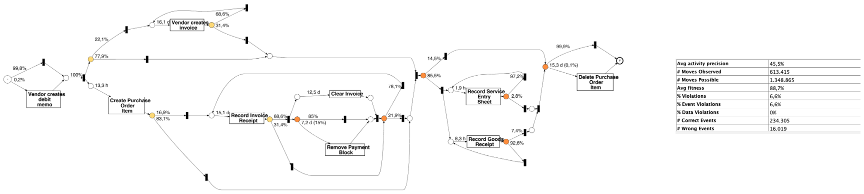
The Model Due to large variety, it is not easy to mine a proper model representing all traces with a high fitness. For simplicity reasons we filter the log for change actvities, since those are analysed individually in section 4. We additionally perform another log split into service orders and non service orders indicated by the occurance of the ”record service entry sheet” event. The two respective models can be found in fig. 27 and fig. 28. The model for non-service goods represents the typical P2P process. Again we see the derivation that a case can end after the creation of an PO Item.



Avg activity precision	100%
# Moves Observed	129.117
# Moves Possible	129.144
Avg fitness	92,1%
% Violations	11,1%
% Event Violations	11,1%
% Data Violations	0%
# Correct Events	43.305
# Wrong Events	5.010

Fig. 27. Model D1: 3-way matching, invoice after goods receipt. Non-Service-PO.

In the second model (fig. 28) 0,2% of all cases start with a debit memo created by the vendor. The other 99,8% skip that step. Concurrently, the vendor can create several invoices while the company creates the purchase order item. After the item is created we can optionally record an invoice receipt which (with an optional payment block removal) is then cleared. After the concurrency optionally we can receive one or more Service Entry Sheets as well as goods. In the end 0,1 % of the PO items is deleted. Note, that the second model got a lower fitness than usual (88%). The precision is only 45,5 %, meaning that it allows for quit a lot more behavior than actually observed in the log. Reason for that is probably the large amount of loops in the model.



Avg activity precision	45,5%
# Moves Observed	613.415
# Moves Possible	1.348.865
Avg fitness	88,7%
% Violations	6,6%
% Event Violations	6,6%
% Data Violations	0%
# Correct Events	234.305
# Wrong Events	16.019

Fig. 28. Model D2: 3-way matching, invoice after goods receipt. Service PO.

Derivations For model D1 we spot quite a lot derivations after the creation of the purchase order. Those are cases where Goods Receipt were missing at this point in the alignment. For model D2 it was not possible to visualize the derivations in Prom 6.

Throughput Time The throughput time of model D1 is visible in fig. 29. On average takes 36 days until an invoice is received and another 31 days until the invoice is then cleared.

The throughput time of model D2 is visible in fig. 28. The waiting time for record invoice receipt is on average 15 days. After that it takes on average 13 days to clear that invoice. If a payment block needs to be removed this takes on average 7 days. The waiting time for record service entry sheet and record goods receipt is surprisingly small. If a PO item is deleted that takes on average 15 days.

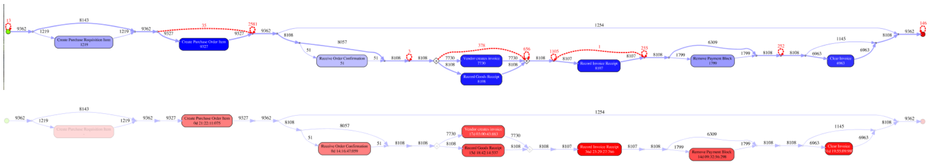


Fig. 29. Case Count and Throughput of Model D.

Order Confirmation Rate Interesting is the order confirmation rate though. Only vendor 433 confirms orders (ca 70% of them in this sublog). All other don't.

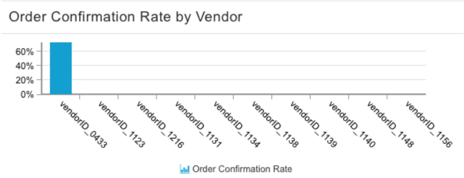


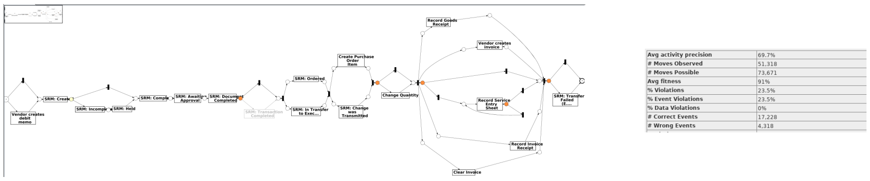
Fig. 30. Order Confirmation Rate of sublog D ordered descending per vendor. Only a subset of all vendors is visible.

2.6 Model E: Supplier Relation Management

We filtered the log so that it only contains traces with SRM Events, within the selected time frame. This log contains 21564 Events with 1425 Cases and 373

Variants of which 2/3 are unique traces. Taking a closer look at the log we see that the attribute `Document Type` has only the "EC Purchase order" value. Further intresting attributes are `Spend classification text` with values "NPR" , "OTHER" and marginally "PR", `Spend area text` with values "Enterprise Services", "CAPEX & SOCS", "Marketing", "Workforce Services", "Others", "Logistics", "Sales", "Real Estate" and "Trading & End Products" in descending case count, `Item Type` which has the values "Service" and "Standard". There is only one `Case-Company` involved, "company_0000". The `Item Category` attribute has "3 way match, invoice after GR" and "3 way match, invoice before GR" values.

We used the inductive miner plug-in from ProM to mine a model for the SRM containing traces which is shown in fig. 31. We used about 89.2% of paths and 100% of activities. This setting assured that we only get one possible loop for the "Record Service Entry Sheet" to maximize fitness and precision of the model. In fig. 31 we present the resulting values regarding fitness and precision of the model which were obtained using the "Multi-Porcess Explorer" ProM plug-in. We observe that there are some places which where marked by the Multi-Process Explorer such as the place after "SRM: Create" or "SRM: Document Complete". While the first one is rather light the latter one is clearly darker which indicates that the precision in this places is between 40% and 60% and therefore lower.



Avg activity precision	69.7%
# Moves Observed	51,318
# Moves Possible	73,671
Avg fitness	91%
% Violations	23.5%
% Event Violations	23.5%
% Data Violations	0%
# Correct Events	17,228
# Wrong Events	4,318

Fig. 31. Spend Classification and Area.

3 Automation Rate

Another KPI we will regard is the automation rate. We use Celonis and the integrated PQL interpreter to realize that KPI. The definition of the KPI can be found in fig. 32. We count the number of events having a user attribute including "batch", denoting an automatically executed event and then divide it by the overall number of events.

```
SUM(CASE WHEN "log.xes"."User" LIKE 'batch' THEN 1.0 ELSE 0.0 END) /
COUNT("log.xes"."Event-Id")
```

Fig. 32. PQL for Automation Rate.

In fig. 33 we can spot the automation rate per activity in the model. We spot at first sight that many activities are always executed manually: All change activities, Clearing Invoices, Block/Delete/Reactvitate PO Items, setting of payment blocks, record of a service entry sheet as well as the creation of debit memos and invoices by the vendors.

On the other hand we also spot a lot evens which are highly automated: Especially most SRM events, except from ”SRM: transaction completed” and ”SRM: Transfer Failed” which both are always executed manually, are highly automated. Other activities are medium to low automated. When we now also take the number of executions into account we realize that especially the already partly automated activities ”Record Goods Receipt”, ”Create Purchase Order Item”, and ”Record Invoice Receipt” could be improved a lot. If possible, automating clearing invoices, recording service entry sheets and the creation of the invoice by the vendor are executed often, though never automatically. In fig. 33 we see the automation of the cases performed by the top 30 vendors in terms of number of items. We see that for the top vendor automation rate is 4% only, while there are also vendors which gain 20% and more. Especially, those vendors which we deal with often offer potential for automation by introducing electronic invoicing.

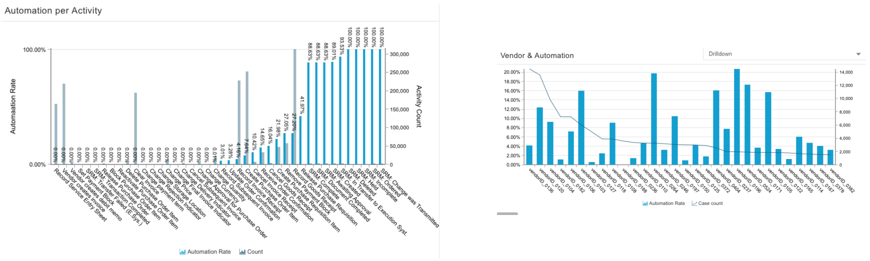


Fig. 33. Automation Rate by activity (left) and Automation Rate by vendor (top 30 vendors) (right)

In fig. 34 we see the automation rate per category. Lowest automation rate is visible in 2-way matches as well as in Consignment cases. three way matches where invoices are received after goods is much more automated than a three way match where the invoices arrives first.

4 Rework Rate

Now, we also want to check the rework rate of the cases. We can define rework in two different ways:

1. Rework as execution of a change activity, like ”Change Price”
2. Rework as multiple executions of certain activties within a trace in a trace.

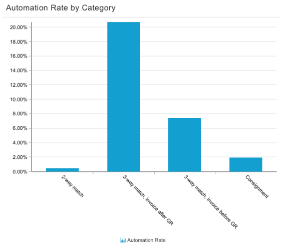


Fig. 34. Automation Rate per category

4.1 Change Activities

We will only regard the first approach in the following. Again we define a KPI for that rework via PQL in Celonis. The query can be found in fig. 35. We compute the relative proportion of all cases containing at least one of the defined rework activities.

```
AVG(CASE WHEN MATCH_ACTIVITIES(NODE_ANY[<%=rework_activities%>] ) = 1 THEN
1.0 ELSE 0.0 END)
```

with Rework Activities:

```
<%=rework_activities%> = 'Change Quantity', 'Change Price', 'Change
Approval for Purchase Order', 'Change Delivery Indicator', 'Change
Storage Location', 'Change Currency', 'Change payment term', 'Change
Rejection Indicator', 'Change Final Invoice Indicator'
```

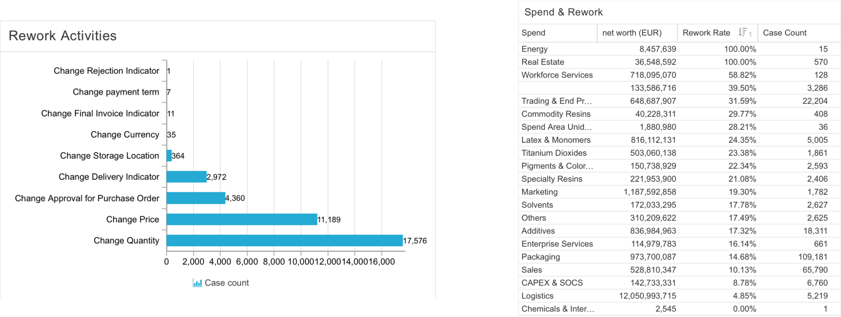
Fig. 35. PQL for Rework Rate.

In fig. 36 we see the number of times each rework event was executed in the filtered log. We observe that especially Change Quantity / Price Approval for Purchase Order / Change Delivery Indicator happen often.

In fig. 36 we also see in which spend area a lot of rework is performed and in which less. Especially "Energy", and "Real Estate" always contain at least one rework activity. These are by the way exactly those fields which exclusively are handled by CompanyID_003. When we dig deeper into that we see that CompanyID_003 in general got a rework rate of 100%. Change Activities are always either "Change Price" (10 times) or more often "Change Approval for PO" (1027). Visible is the low rework rate for large areas like Packing and Sales. Critical is the rework rate of 33% in "Trading and Endproducts" while the the number of cases is large (22204 cases).

Let us now look at the rework rate per vendor. In fig. 37 we see the top 25 vendors in terms of net value taken from the purchase order creation (left) and case count (right). Critical vendors are marked in orange and red.

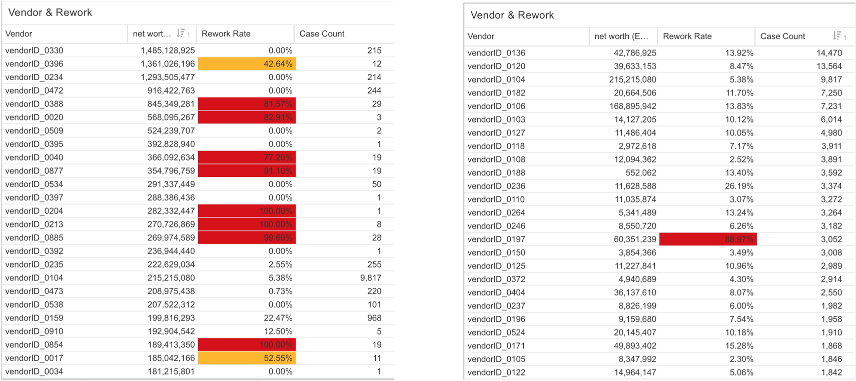
In figure fig. 38 we now spot the rework rate per category.



Spend & Rework

Spend	net worth (EUR)	Rework Rate	Case Count
Energy	8,457,639	100.00%	15
Real Estate	36,548,592	100.00%	570
Workforce Services	718,095,070	58.82%	128
	133,586,716	39.50%	3,286
Trading & End Pr…	648,687,907	31.59%	22,204
Commodity Resins	40,228,311	29.77%	408
Spend Area Unid…	1,880,980	28.21%	36
Latex & Monomers	816,112,131	24.35%	5,005
Titanium Dioxides	503,060,138	23.38%	1,861
Pigments & Color…	150,738,929	22.34%	2,593
Specialty Resins	221,953,900	21.08%	2,406
Marketing	1,187,592,858	19.30%	1,782
Solvents	172,033,295	17.78%	2,627
Others	310,209,622	17.49%	2,625
Additives	836,984,963	17.32%	18,311
Enterprise Services	114,979,783	16.14%	661
Packaging	973,700,087	14.68%	109,181
Sales	528,810,347	10.13%	65,790
CAPEX & SOCS	142,733,331	8.78%	6,760
Logistics	12,050,993,715	4.85%	5,219
Chemicals & Inter…	2,545	0.00%	1

Fig. 36. Rework Activities (left) and Rework Rate by spend area (right)



Vendor & Rework

Vendor	net wort…	Rework Rate	Case Count
vendorID_0330	1,485,128,925	0.00%	215
vendorID_0396	1,361,026,196	42.64%	12
vendorID_0234	1,293,505,477	0.00%	214
vendorID_0472	916,422,763	0.00%	244
vendorID_0388	845,349,281	81.57%	29
vendorID_0020	568,095,267	82.91%	3
vendorID_0509	524,239,707	0.00%	2
vendorID_0395	392,828,940	0.00%	1
vendorID_0040	366,092,634	77.20%	19
vendorID_0877	354,796,759	91.10%	19
vendorID_0534	291,337,449	0.00%	50
vendorID_0397	288,386,436	0.00%	1
vendorID_0204	282,332,447	100.00%	1
vendorID_0213	270,726,869	100.00%	8
vendorID_0885	269,974,589	99.69%	28
vendorID_0392	236,944,440	0.00%	1
vendorID_0235	222,629,034	2.55%	255
vendorID_0104	215,215,080	5.38%	9,817
vendorID_0473	208,975,438	0.73%	220
vendorID_0538	207,522,312	0.00%	101
vendorID_0159	199,816,293	22.47%	968
vendorID_0910	192,904,542	12.50%	5
vendorID_0854	189,413,350	100.00%	19
vendorID_0017	185,042,166	52.55%	11
vendorID_0034	181,215,801	0.00%	1

Vendor & Rework

Vendor	net worth (E…	Rework Rate	Case Count
vendorID_0136	42,786,925	13.92%	14,470
vendorID_0120	39,633,153	8.47%	13,564
vendorID_0104	215,215,080	5.38%	9,817
vendorID_0182	20,664,506	11.70%	7,250
vendorID_0106	168,895,942	13.83%	7,231
vendorID_0103	14,127,205	10.12%	6,014
vendorID_0127	11,486,404	10.05%	4,980
vendorID_0118	2,972,618	7.17%	3,911
vendorID_0108	12,094,362	2.52%	3,891
vendorID_0188	552,062	13.40%	3,592
vendorID_0236	11,628,588	26.19%	3,374
vendorID_0110	11,035,874	3.07%	3,272
vendorID_0264	5,341,489	13.24%	3,264
vendorID_0246	8,550,720	6.26%	3,182
vendorID_0197	60,351,239	88.97%	3,052
vendorID_0150	3,854,366	3.49%	3,008
vendorID_0125	11,227,841	10.96%	2,989
vendorID_0372	4,940,689	4.30%	2,914
vendorID_0404	36,137,610	8.07%	2,550
vendorID_0237	8,826,199	6.00%	1,982
vendorID_0196	9,159,680	7.54%	1,958
vendorID_0524	20,145,407	10.18%	1,910
vendorID_0171	49,893,402	15.28%	1,868
vendorID_0105	8,347,992	2.30%	1,846
vendorID_0122	14,964,147	5.06%	1,842

Fig. 37. Rework top 25 vendors (value) (left) and Rework top 25 (case count) (right)

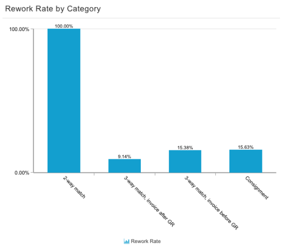


Fig. 38. Rework Rate per category

Notable is the 100% rework rate for 2 way matches. Reason for that is the obligatory execution of "Change Approval for Purchase Order Item" as visible in fig. 13. In three way matches where invoices arrived before goods we observe a slightly higher rework rate than in the case where goods arrived before the invoice.

5 Social Mining

First we will present general social facts about the log. The log contains two kinds of resources: users and batch. The difference between these types of resources is, as described in the challenge description, that "batch" resources are automatically executed while "user" resources are actually manual resources. As we can't really distinguish between the 20 batch users we decided that we aggregate all batch resources as one "batch" resource. After this manipulation we got 608 resources. Using the PM4PY modules we discovered that there are events where no resource was recorded, these are ['Vendor creates invoice', 'Record Service Entry Sheet', 'Vendor creates debit memo', 'Clear Invoice']. Batch resources execute the following tasks: ['SRM: Created', 'SRM: Complete', 'SRM: Awaiting Approval', 'SRM: Document Completed', 'SRM: Ordered', 'SRM: In Transfer to Execution Syst.', 'SRM: Change was Transmitted', 'Record Invoice Receipt', 'Create Purchase Order Item', 'Create Purchase Requisition Item', 'Remove Payment Block', 'Receive Order Confirmation', 'Update Order Confirmation', 'Record Goods Receipt', 'Record Subsequent Invoice', 'Cancel Invoice Receipt', 'SRM: Deleted', 'Cancel Goods Receipt', 'SRM: Incomplete', 'SRM: Held', 'Change Quantity', 'Release Purchase Requisition']. We have 288 users that only execute one task, with 123 users doing only 'Create Purchase Requisition Item'. In the next step we decided to filter out these 288 users to be able to progress with our Social Network Analysis, as the overall resource count was to high to be analysed properly. Additionally, we found that in 87355 traces the respective resource only executed the activities ['Record Goods Receipt', 'Cancel Goods Receipt'], not necessarily one time only but for sure only these two activities. Therefore, in order to be able to cluster resources by their activities, we renamed these resources to "rcgr_users". One thing that we noticed throughout the different social analysis is that 'user_297' always appears to be isolated. Looking closely this resource only performs two tasks 'Create Purchase Order Item' and 'Delete Purchase Order Item' on one specific day (31.01.2018). The 'Create Purchase Order Item' is performed four times consecutively with four different values and 'company_0000'. 2 minutes later 'Delete Purchase Order Item' is performed four times with the same values and company. 'user_013' performs the most activities as can be seen in the following figures.

6 Summary

In this report we analyzed the given log first in general and saw anomalies in company 1 and 2 which each only execute one PO Item in 2018 which was both deleted. We encountered a highly complex process with many variations. By splitting it according to the categories "2 way matches", "Consignments", "3way matches, invoice after goods" and "3 way matches, invoices before goods" we could get better insights. Then, models for each of those sublogs were created, each with sufficiently high fitness. Via a throughput analysis we found out that the time between reporting of the invoice receipt and the clearance is high, same holds for the time between record goods receipt and record invoice receipt. On some sublogs we computed the order confirmation rate. Other KPIs we observed were the automation rate where we spotted which activities are well automated and which offer potential for automation. Last but not least we looked into the rework rate and saw that especially often prices and quality changes were executed. For all KPIs we analyzed what vendor or resource was particularly often involved in not well working cases, so that improvements can be executed based on this analysis. For instance did we observe a high amount of one-time-vendors.

References

1. Bpi challenge 2019. https://icpmconference.org/icpm-2019/contests-challenges/bpi-challenge-2019/. Accessed: 2019-05-24.
2. An introduction to purchase to pay process flow. https://www.purchasecontrol.com/uk/blog/procure-to-pay-cycle/. Accessed: 2019-05-24.
3. van dongen, b.f., dataset bpi challenge 2019. 4tu.centre for research data. https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1.