

# Business Process Intelligence Challenge 2019 – a contribute

Lorenzo Botti

lorenzo.botti@logtovalue.com

## Abstract

Process mining technology is there but most businesses are still wary of its adoption, because many data-based insights are made available by business intelligence. This contribute aims to show how both approaches are needed and how process-insights can improve decision making and representation of business performance through KPIs. This Challenge is a useful practice to highlight process mining advantages, but it is not a process mining project, so many assumptions will be left unchecked and more questions will be raised than answered given the lack of confrontation with the Client.

## 1 Introduction

The available dataset\* is a collection of cases<sup>†</sup> from a Purchase-to-Pay process, that take place - or at least begin - in 2018<sup>‡</sup>. A thorough clean-up and then filtering out of cases is required.

A critical approach about the dataset is followed in order to analyze both visible information (cases, activities etc.) and the set of information that lies behind (assumptions on ETL phase and on the real-life daily routine of the process), as well as “noise” in the event-log.

The analysis encompasses business-intelligence style analytics and process mining, so to show at what extent both approaches can thrive from a synergy between them, and how businesses can fact-check their KPIs set with process-based evidences. While pursuing a suitable set of process models for the most part of traces, peculiar behavior of subsets will be noted and discussed.

For the purpose of the analysis and to the best knowledge of the dataset, it is assumed that data had been collected from an ERP-like system (or a multiplicity of systems) on which both internal users and external ones – vendors - can operate. The latter ones either directly or intermediated by back-office staff.

---

\* van Dongen, B.F., Dataset BPI Challenge 2019. 4TU.Centre for Research Data.

<https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>

<sup>†</sup> in the text terms “case”, “process instance” and “trace” and occasionally also “sequence” will address the same meaning

<sup>‡</sup> <https://icpmconference.org/icpm-2019/contests-challenges/bpi-challenge-2019/>

## 2 A look at the dataset

A first look at data shows some peculiar features that call for a clean-up. It is however necessary to retain all the useful information and potential insights.

### 2.1 Calendar

The dataset refers to a Purchase to Pay process taking place in 2018, that is beginning in 2018. Last events should be assumed no further that date of dataset creation (January 2019).

Some activities have been recorded with incongruent calendar dates, spanning from 1948 to 2017, and beyond January 2019 until April 2020 (Table 1).

Traces	251.734
Events	1.595.923
First Event	1948-01-26T23:59:00Z
Last Event	2020-04-09T23:59:00Z

**Table 1:** original dataset

These activities are almost uniquely related to vendor actions (Vendor creates debit memo, Vendor creates invoice). Just few are due to “Create Purchase” activities that take place in 2017.

It is possible to make two hypotheses: a deliberate or careless behavior on one side, an IT-related glitch on the other side (maybe due to differences in machine set-up of dates and times across different terminals or user interfaces). A third, most probable, hypothesis – induced noise in the event log – cannot be either excluded either investigated, so just the other two are considered.

As for the first hypothesis, involved vendors also participate to other cases where no calendar issue takes place, so it is possible to say that they are aware of how to process information and upload them into the system. For the same reason, not even the terminal set-ups can be associated to the issue. Besides, Vendor-related activities (“Vendor creates invoice” and “Vendor creates debit memo”) are all performed only by an user identified as “NONE” and always at 23:59; that is, similar activities related to different instances of the process are uploaded as a batch operation onto the “ERP-like” system at the cut-off time of the day (it is an usual feature of multisite ERP-like systems to go through a periodic planned update of all the tables associated with the underlying database). The fact that even the “out of calendar” vendor activities had been performed at 23:59 tells that either the system failed on recording the date of execution (but not the time) or that the associated date is an invoice-based input and not a system one. The latter hypothesis is not sound with any practice of ERP architecture so only the first one is to be considered. Therefore, an IT-glitch is probably to be investigated for the “out of calendar” events; this glitch happened either at “run time” (when data are recorded by the ERP-like system) or even at the time of data extraction and transformation (the ETL phase for preparation of event log).

The amount of cases is about 0.1 percent (271 line-items) and no recurrent features can be found (spend area, company etc.). It is therefore possible to clean out these cases without fear of any information loss toward the process itself. Few instances originated in the last month of 2017 and then completed the process in the first months of 2018, but their small number still allows for them being excluded by the dataset.

Note that very few of those cases (7 out of 271) are due to “create purchase requisition item” (therefore executed by internal users) and span along the year 2017. It could be the case of an “error” or a business accepted process drift (creating the document “in the system” in the afterward of other activities that should be consequent to the document itself)<sup>§</sup>.

---

<sup>§</sup> If this is true, nonetheless any user interface should have a control check on dates associated to an event, with a specific procedure when an input date is different from the current (machine) one.

## 2.2 Follow the money

An important feature stands out when looking at the very essence of any Purchase to Pay process: what are we paying for? Cases are classified with three attributes that refer to Spend classification categories (Classification, Area text, Sub Spend Area Text). Among the “Area text” there is an “Spend Area Unidentified” to deal with unforeseen or unchecked spend areas (36 traces belong to this classification). It is surprising to see though (Figure 1) that there are 3289 traces associated to a *blank* classification. Not only there is no classification, but at a deeper analysis no peculiar feature can be associated with these cases (vendor, user, item category etc.) apart that 98% of them are referred to as a “Standard” Item Type (which are anyway the most common traces in the set, ca 90%).

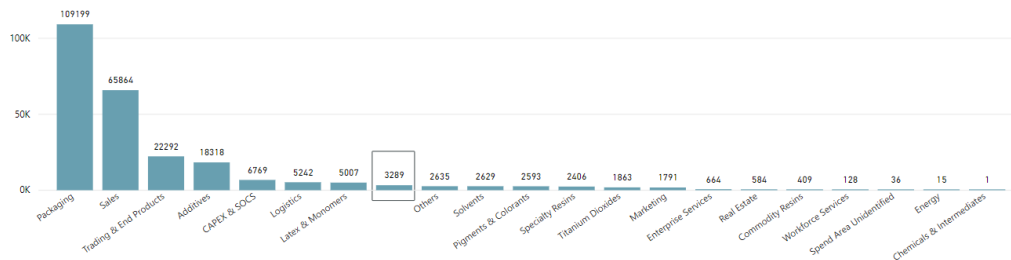


Figure 1 Number of traces associated to a Spend Area

Most of these anonymous traces are “complete” (for 1622 traces there is at least one “Clear invoice” activity). No actionable insight can be found or assumed at this point. If it is not due to a deliberate action (e.g.: to introduce noise in the event-log), then it would be matter for an accounting audit\*\*. These cases are better excluded from a first analysis†† even though other attributes are available to fit those traces into different process categories (Item Category, Item Type etc.).

## 3 The working set, from intelligence to mining

The clean-up leaves us with a dataset built on 74.851 Purchasing Documents (Vs 76.349) and 248.177 cases (Vs 251.734). The working dataset will be subject to further filtering, but as a first step, a look at the dimensions into which the dataset can be broken down already provides some useful evidence. What follows is a just a partial business intelligence analysis, that can be extended and is surely summed up in dashboards and reports. The focus here is to show how some features can lead us toward a process mining approach. This will support the mining itself and will help to verify assumptions.

Process mining relies on timestamps, the date and time reference of every activity. In the dataset there are some activities, within the same trace, that share unique timestamps. The fact here is not about the precision of the reference (seconds, not microseconds), but to the reality of a “stamp”. Activities within a trace that share the same timestamps are to be assumed as executed by “ticking boxes” on a user interface screen-view (or by a batch-user) and then “time-stamped” at the moment of execution of the screen-view (eg.: “save”, “next page”...). In such hypothesis, the sequence of these

\*\* A rough and inaccurate estimate of the value involved can be given with two figures: the EUR amount of the sum of “Create Purchase Order” is about 25 EUR Million, while the sum of “Clear Invoice” is about 15 EUR Million.

†† After the clean-up, the count of unique users will decrease by 6 units (out of 627) and that of unique vendors will decrease by 45 (out of 1975); these are anyway just a tiny portion of those involved in the wiped-out cases (262 users, 183 vendors), so no specific actions on user training or vendor contract management would address the situation.

events will be recorded as per process-design and can help explore assumed model (those developed by IT according to business requirements, at least), even if for limited number of activities (such is the case of SRM-likes that belong to E.C. Purchase Order Document Type, and are present in both “3 way Before” and “3 way After” dataflows). Other activities simply share the same timestamp because they are repetitions and related traces will need a deeper investigation in order to understand if those repetitions are due or not in the process.

### 3.1 Companies, organization and users

The available dataset is supposedly the collection of cases from a Parent Company headquartered in the Netherlands and some of its sixty subsidiaries worldwide. No information is available on the organization, but process instances can reveal some insights.

Traces are associated to four different Companies (Table 2). Most of traces are operated within CompanyID000, while CompanyID\_003 shows a specialization (“2\_way\_match”).

	CompanyID000	CompanyID001	CompanyID002	CompanyID003
2_way_match	-	-	-	995
3_way_match_after	14831	2	-	-
3_way_match_before	217874	-	2	-
Consignment	14473	-	-	-

**Table 2** Distribution of cases by Companies and Item Category

CompanyID000 is associated to over 99% of instances. Either the Company is an aggregation of some subsidiaries or represent sort of Business Unit (by this approach CompanyID001 and CompanyID002 could be considered “noise”). By breaking down through the Spend Area attribute of the events, CompanyID003 is probably a Service Company to other subsidiaries: in fact, “energy” related purchases are exclusive to this company, just like most of “real estate” ones.

Focusing on users, CompanyID003 has 19 users, among which 4 are exclusive to the company (602, 603, 604, 606) and three of these show a clear specialization: 604 for clearing invoices, 603 in conjunction with 602 for creating and approving Purchase Orders. The relationship between the latter two users (603 and 602) will also have an impact on the discoverable process model, mostly due to the different meaning of an activity they both will perform (*Change approval for purchase order*).

The other companies share 618 users. A sub aggregation of CompanyID000 is still possible through a social analysis based on *handover of work*. In fact, working on “consignment” traces there are three sub-groups of users (out of 155 users) that never interact among each other. Consignment traces have the peculiarity that users probably operate at local companies’ plant/warehouse locations (Goods-related activities); besides, these users, even in other Item-category traces (3 way After, 3 way Before) execute only the same set of activities. In the end, the three sub-groups could refer to the number of locations or subsidiaries involved<sup>‡‡</sup>.

### 3.2 Cumulative Net Worth (EUR), matching activities and data check

The dataset provides a “Cumulative net worth” figure for every activity within any process instance<sup>§§</sup>. There can be the case of multiple invoices and good receipts within the same trace, but they all have the same cumulative net worth, which indeed is a cumulative figure not an atomic one; for example, the twelve *goods receipts* of trace 2000000055\_01 share the same value (43,312.00 EUR) and this mean that each one would have been on average 3,609.00 EUR. This is verifiable for those spend areas for which multiple invoices are expected (workforce, real estate etc.). The matter is

<sup>‡‡</sup> “Consignment” users do not take part to any “2 way match” traces, as expected, being Goods activities not included in these traces; this confirm “specialization” of these users and the peculiar nature of “2 way match”/CompanyID02

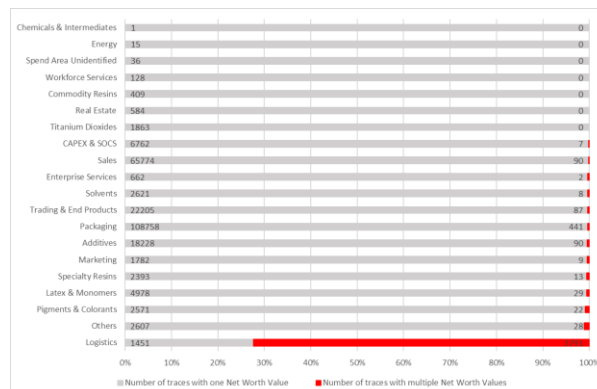
<sup>§§</sup> Exception made for “consignment” traces

that each trace should have a unique value of cumulative net worth, as it happens indeed for 98% of traces (Table 3). The compliance assumption is that activities should be matched against the “cumulative net worth” figure (namely, *good receipt* and *invoices*), but for 98% of traces it is only possible to *count* activities not to *match* their values. Looking at the residual 2% of traces with more than one EUR value it is possible to question which activities are matched within the process and raise some questions about event-log preparation and user interface controls.

Total traces (original dataset), units	251.734
Traces with one unique value associated to the process instance (units)	247.102
Traces with 2 to 75 values associated to the process instance, (units)	4.632

**Table 3** number of process instances aggregated toward the numerosity of EUR figures through the instance

First, we look *where* traces with multiple EUR values occur, and it is clear they are mostly confined in specific spend areas (Figure 2) and fall into “3 way After” (86%) and “3 way Before” (14%) categories and over 82% are related to *Service Item Type*.



**Figure 2 – Number of traces with one or multiple Net Worth values (across Spend Area)**

Looking at the process, there are many different sequences and patterns (for example: Figure 3) that require some questioning. Given that the reference is a dataset originated from an ERP-like database, figures could have been affected by data transformation (were original database figures atomic or cumulative?), and matching activities could have been impacted.

_case_concept_name_	_event_concept_name_	_event_time_timestamp_	_event_Cumulative_net_worth_EUR_	_event_User_
4507006350_00001	Create Purchase Order Item	30/01/2018 13:51:00	276.00	user_200
4507006350_00001	Record Goods Receipt	30/01/2018 13:51:00	552.00	user_200
4507006350_00001	Record Service Entry Sheet	30/01/2018 13:51:00	276.00	NONE
4507006350_00001	Record Goods Receipt	31/01/2018 11:19:00	18.798.00	user_200
4507006350_00001	Record Service Entry Sheet	31/01/2018 11:19:00	276.00	NONE
4507006350_00001	Vendor creates invoice	06/02/2018 23:59:00	276.00	NONE
4507006350_00001	Record Invoice Receipt	12/03/2018 17:36:00	552.00	user_001
4507006350_00001	Clear Invoice	29/03/2018 15:11:00	276.00	user_002

**Figure 3 – multiple values within a trace, a case of anomaly for “Record Goods receipt”**

“Goods receipt” value apparently is *independent* from other activities, while value matching activities would fall in two distinct sets: first one with “Create P.O.”/“Vendor Creates Invoice”/“Clear Invoice”, and second one with “Record Service Entry sheet”/“Record Invoice receipt” (the Invoice receipt value being the sum of Service sheet entries). In the end some questions can be raised about: the origin of input for “goods receipt”, the reason for manifold and instantaneous repetitions of both “Record service entry sheet” and “Record Goods receipt”, the input for matching existing invoices and related “Clear Invoice” activity.

### 3.3 Dataflows, documents and items: dimensions for modeling

Process instances can follow at least four main data flows and complexity grows according to Document Type and Item Type (Figure 4). While Item Categories, as described, allow some speculation about underlying process model, there are useful insights based on the other two dimensions. For example, some set of activities are specific to document type (*SRM-...* for E.C. Purchase Order). Another example is that some *Document type* traces (Standard P.O.) show a process drift in the period. The complexity of “Service” traces only belongs to “3 way After”, as well as unique Item Type and Document Type are peculiar for both “Consignment” and “2 way match” data-flows.

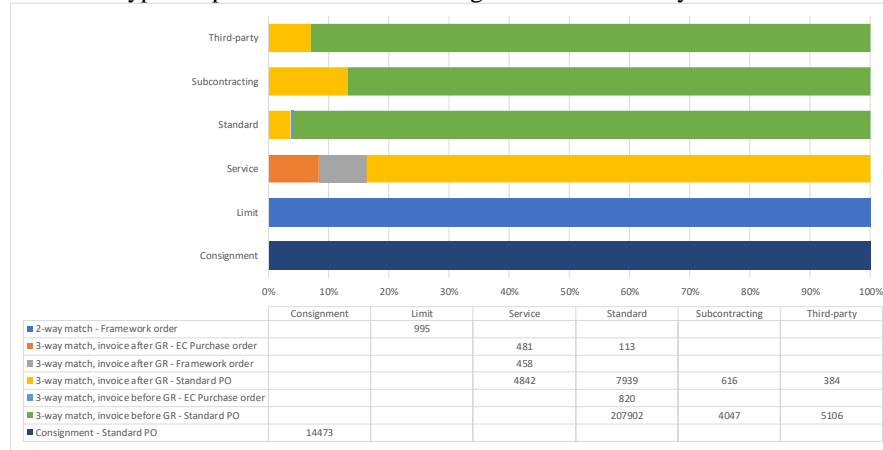


Figure 4 – Distribution of traces across Item Type, Document Type and Item Category

### 3.4 Users and activities, pattern anomaly

There are over six hundred users that take part to the process in the period and about 1.5 million activities. As shown in Figure 5, there is a sharp increase of users in the last months of the year. From an average of 281 users between January and August, the count passes to an average of 396 users between September and December. This increase is apparently not related to an increase of activities performed; by breaking down these figures to Item Categories, the sharp increase in number of users is clearly linked to traces associated with a “3 way Before” data flow and particularly with traces associated to “Standard P.O.” document type (*see* 3.3). These are the same traces that will show a higher not-compliance, so it would be interesting to discuss if more resources had been spent to deal with “complexity” or if the more resources involved were the cause of such complexity due to training issues.

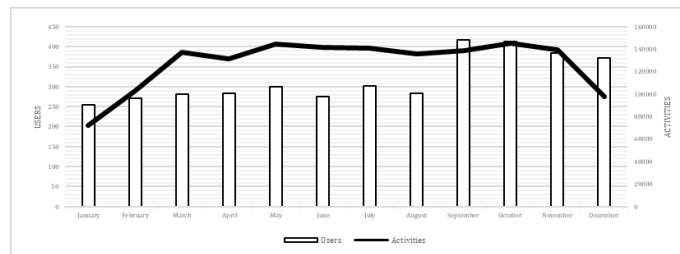


Figure 5: users and activities per month (2018)

Besides, looking at the process - *who does what and when* along the instance - there are some peculiarities with some (1401) of the traces that belong to a subset with a “debit memo” event.

In brief they show a pattern as in in Figure 6 (highlighted and boxed part).

12/01/2018 07:33:00	user_038	Create Purchase Order Item
12/01/2018 23:59:00	NONE	Vendor creates debit memo
12/01/2018 23:59:00	NONE	Vendor creates invoice
23/01/2018 16:35:00	user_007	Record Invoice Receipt
23/01/2018 17:39:00	user_029	Record Goods Receipt
24/01/2018 09:40:00	user_006	Cancel Invoice Receipt
24/01/2018 09:41:00	user_006	Clear Invoice
24/01/2018 09:43:00	user_006	Record Invoice Receipt
01/02/2018 12:01:00	user_002	Clear Invoice

Figure 6 – peculiar pattern on Invoice

The box-highlighted pattern (same user, anomaly sequence in few minutes) occurs in any of the potential dataflows (obviously not for *Consignment* traces). The role of Debit memo would be an interesting matter of discussion with management and IT department<sup>\*\*\*</sup>. Indeed, any *Debit Memo* is followed by *invoice* related activities (*cancel*, *clear* etc.), with no helpful clues to discriminate these (debit memo) activities from the ones related to “actual” invoices. The fact that boxed activities in Figure 6 are executed by the same user in matter of minutes implies a probable business rule that would need attention.

A smaller subset (377 traces) shows a similar pattern with Goods related activities (Figure 7). In this case, the time interval stretches from minutes to days, but still the same user performs the activities.

Activity	Resource	Date	Time
Create Purchase Order Item	user_079	03.01.2018	15:14:00
Vendor creates invoice	NONE	22.01.2018	23:59:00
Record Goods Receipt	user_094	23.01.2018	12:09:00
Cancel Goods Receipt	user_094	30.01.2018	08:14:00
Record Goods Receipt	user_094	30.01.2018	13:06:00
Record Invoice Receipt	user_015	19.07.2018	09:32:00
Clear Invoice	user_002	19.07.2018	13:16:00

Figure 7 – peculiar pattern on Goods receipt

Some assumptions<sup>†††</sup> can be made to explain the above patterns (operational mistake, intent etc.) and a deeper investigation with the Client is needed. Still, a look at the handling of Logistics P.O. (road packed) is advisable (three case\_vendors cover 44% of traces: 534, 388, 472).

### 3.5 Process drift and forced execution

A process can change along a period for many reasons: workload, seasonality, business changes, etc... When mining process data, a drift is then spotted. In the dataset, at least two changes apparently took place in the process and they can be categorized as it follows:

- Activity-based process change (a change in sequence)
- Resource-based process change (a change in resource for specific activities)

<sup>\*\*\*</sup> According to accounting practices and evidences in the dataset, a debit memo – formally raised by a Vendor in this case – can be triggered either by the Company or by the Vendor. In the latter case, it acts as an incremental invoicing or a note that there are unpaid invoices/expenses; a Company can trigger it directly or indirectly, as it is with some instances that show “debit memo” occurring after some Goods-adjustment (cancel, change price etc) even in absence of an invoice, probably meaning that the memo is related to a part of expenses (freight...) owed to the Vendor even if goods have been refused (etc.). Cases can be multiple, and they all are supported by the dataset. The handling of any instance could improve by using different names (Cancel Memo/Cancel Invoice, Clear memo/Clear Invoice...) with the advantage of process compliance and KPIs, at the expense of few more tables and relations in the database.

<sup>†††</sup> For cases with adequate time interval, similar behavior can occur in processes where there is a performance-based reward system in place and some activities are performed just to “re-start” the clock; or it could be a sign of a work in progress audit on goods receipt. Where time interval is just a matter of minute, it could be an operational mistake, the cause of which deserve attention

As for the first one, apparently there has been a change in the start events of traces within the category of Standard P.O. document type (Figure 8) between Jan-Aug and Sept-Dec. Within the two sub-periods the total number of Standard P.O. traces are almost equal, but the difference in start events of traces show a change in process. It is certainly not a matter of resource training and it is not associated to specific vendor or item. The scale of the phenomenon can be a consequence of a business decision (procurement mgmt. to monitor the creation of new P.O., Organization acting on performance-based metrics, etc.) or even related to technical requirements (a change in database structure or administration). A discussion with management could help to justify the drift.

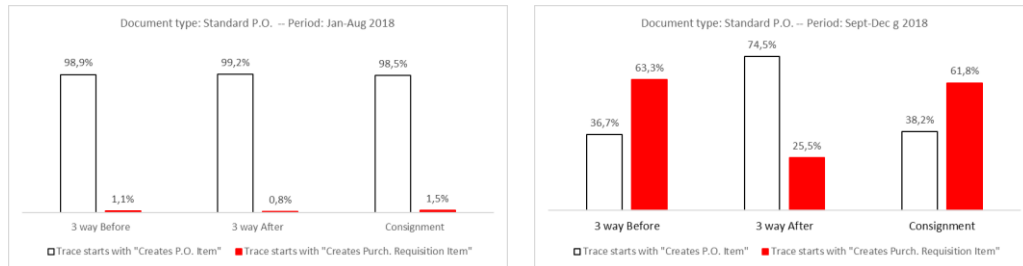


Figure 8 – change in start events distribution along the period

As for the resource-based process change, something peculiar happened with “Clear Invoice” on a specific date. It is not a real process drift and it is better described as a forced action, probably executed in order to “close” cases. If so, it tells that system can be forced and this should happen only under extraordinary circumstances - and under strict supervision.

“Clear invoice” activity can be interpreted as an authorization, not the material payment action, which is probably done by accounting department in a subsequent process. Even as such, “Clear invoice” is mostly executed by few resources with specific authorization profile within the organization. Indeed, for “2 way match” 84% of “Clear invoice” are executed by the same resource (604), whereas for both “3 way Before” and “3 way After” on average over 85% of times “Clear Invoice” is executed by just one resource (002). These “Clear invoice” events happen *in batches*, and this would fit with the hypothesis of a “middle-manager” that periodically releases cases worked-on by others (a typical arrangement of segregation of duties), by *approving* the “Clear Invoice”.

Overall, this is constant across the entire period. Then, at the beginning of January 2019 (between Jan 2<sup>nd</sup> and 7<sup>th</sup>, with a peak on the 4<sup>th</sup>), a subset of 7868 traces end with a “Clear invoice” activity executed by “None”<sup>†††</sup>. In brief, a batch of thousands of process instances supposedly have been forcefully “cleared”. The reason is understandable, but these extraordinary measures should be carefully monitored and properly authorized by management.

## 4 Process models, trace completeness and compliance

Before exploring compliant traces, it is worth mentioning a subset of 1410 traces which all end with a “Record Invoice receipt” activity, but do not show any prior “Vendor creates Invoice/Debit memo”. These traces are not complete – no “Clear invoice” – so formally they can’t possibly be categorized as not-compliant, nonetheless, the absence of any “Creates invoice” within these traces shows a lack of compliance related to expected sequence. These traces belong to *Sub-contracting* (45%) and *Standard* (42%) item type and what is more important, they have the “Record invoice receipt” activity executed 90% of times by “batch01”, while in the entire set, this activity is usually

<sup>†††</sup> During the same period over nine thousand traces end with the same activity executed by actual users (not by “None”).



executed (over 90%) by *user\_X* (human users). Thus, it could be a case of process change (as in 3.5 *Process drift and forced execution*) to be investigated, at least because the last event take place months after the second-to-last event.

The dataset is a snapshot of what happened in a certain period, and as such it is a collection of process instances, either partial or complete. For the discovery of process model(s) only complete traces will be investigated. Assumptions are needed about the meaning of “complete”.

The minimal assumption for completeness is about a set of mandatory activities, which can be grouped according to dataflows.

3 way Before	3 way After	2 way	Consignment
Create P.O.	Create P.O.	Create P.O.	Create P.O.
Record Goods Receipt	Record good Receipt		Record Goods receipt
Vendor Creates Invoice	Vendor Creates Invoice	Vendor Creates Invoice	
Record Invoice Receipt	Record Invoice Receipt	Record Invoice receipt	
(Remove Payment Block)			
Clear Invoice	Clear Invoice	Clear Invoice	

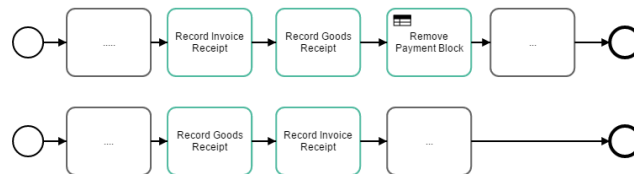
**Table 4 – Mandatory activities for traces “completeness” according to dataflow (not in sequence order)**

Focusing on traces with a minimum set of activities for trace completeness (Table 4), then, compliance must be investigated, along two dimensions: process-compliance and EUR-value compliance.

As per the latter, we have seen (Figure 3) that about a third of “Logistics” traces could be apparently not-compliant based on Goods Receipt EUR value; they are anyway compliant (through “Record Service entry sheet”, thus raising questions on where the EUR-value information come from). The total number of these traces allow for them to be excluded from compliance analysis.

Process-compliance can be just about the presence of mandatory activities or even about an expected sequence, with different impact through different dataflows.

The most critical path (and with most of traces in the period) is the “3 way Before”. Process instances associated to this dataflow belong to two different Document Types (E.C. Purchase Order, Standard P.O.).



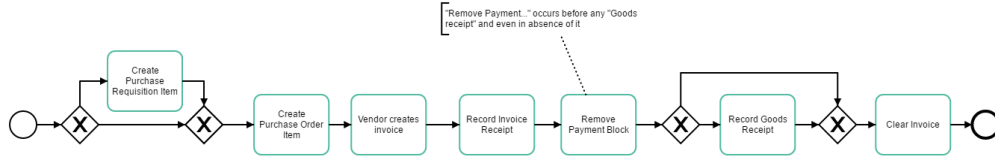
**Figure 9 – “3 way Before” traces, process-compliance basic sequences**

As in Figure 9, for these traces to be process-compliant there are two chances: if a “Record Invoice Receipt” precedes a “Record Good Receipt”, then a “Remove Payment Block” must follow, while the latter is not needed if “Record Goods receipt” precedes “Record Invoice Receipt”. Besides, if required by the sequence (as in upper part of Figure 9), “Remove Payment block” must precede any invoice clearing.

Based on these assumptions, not-compliant instances (Figure 10), even if present in small numbers, show severe violations, caused by:

- a lack of business rule (the system neither imposes a strict sequence-compliance neither requires mandatory activities)

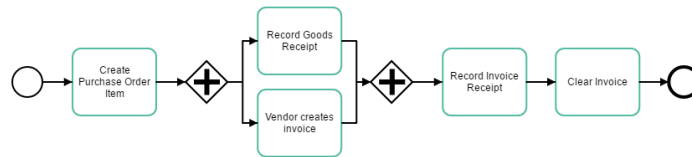
- a lack of monitoring (segregation of duties should apply, preventing any invoice-clearing in violation of rules).



**Figure 10 – “3 way before” not compliant traces (358 traces, with 77 traces without “Record Goods receipt”)**

It’s worth mentioning that “Remove Payment Block” is present also in a subset of traces where “Record Goods receipt” precedes (lower part of Figure 9) “Record Invoice Receipt” (it happens in about 30 thousand cases, while it doesn’t in about 120 thousand cases). It would be interesting to discuss which business rules apply and if there are more than one: is Remove Payment required or not only according to the sequence (Figure 9), or do other conditional rules apply (vendor, items, EUR-values...)? In case of a set of rules, is there a hierarchy?

As for “3 way After” traces, process compliance requires the presence of few mandatory activities, and with no strict sequence rules (the one in Figure 11 only cover 40% of *invoice-cleared* process instances).



**Figure 11 – “3 way After” traces, minimum set of mandatory activities (weak assumption on sequence)**

This category of traces mostly shows compliance within the full subset (Table 5).

3 way After	Compliant	Compliant subset	Not Compliant
Standard P.O.	8645	1885	(149)
E.C. Purch. Order	436	20	-
Framework Order	339	56	-

**Table 5 – Compliance of traces based on activities and EUR values**

There is a subset of compliant traces that include an unexpected activity: “Remove payment block”. All traces have the GR-based set at *true*. If this is correct, then, such a subset would require some questioning about the rules behind the payment block (which must be different from that in “3 way Before” instances). On the other hand, the GR-based value could have been wrongly set (if it were *false*, those instances would belong to “3 way Before” and would be fully compliant).

Few traces are not compliant based on Eur-value, but the statement must be proved. It is already known (Figure 3) that “Record Goods Receipt” can differ in values as well as “Record Invoice receipt”. On the other hand, “Create P.O.” and “Vendor creates invoice” have same EUR value of “Clear Invoice”. In the end, value-based compliance is full if referred only to EUR value of Vendor Creates Invoice, Clear Invoice and Create P.O., and mostly full if also Goods receipt and Invoice receipt are considered. The anomalies should be investigated though, and questions be raised about which is the actual input or information that “tells” the operator which EUR values are associated to Goods Receipt and Invoice Receipt.

As for traces that follow a “2 way-match” dataflow, the complete ones (286) apparently are compliant based on presence (not sequence) of activities and EUR values. A process perspective aims also to look at *who does what* and this will reveal a probable *four-eyes principle* in the set. Most of traces are incomplete, even if active since the first months of 2018. This is in line with a Framework Order document type (100% of 2 way match traces), but still could be a matter for audit, even if it happens that Year1 purchase orders are cleared at the beginning of Year2 (dataset was made available on Jan. 19<sup>th</sup>).

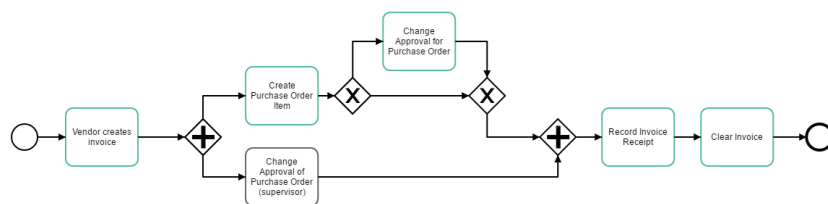
Consignment traces are mostly complete (13456) and fully compliant. They are mostly affected by the change in process (see 3.5) at the start of any instance, but this doesn’t impact on the model, which is actually a basic one.

## 5 Process models

Models are investigated for most common set of traces within different dataflows (Figure 4) and main variants, thus, for example, different Start events will be considered only if they have a strong impact on the sequence. Peculiar behavior already mentioned in the text (debit memo, same-user activities etc.) is not included in the analysis, which aims to general schemas or subprocesses.

## 6 “2 way match – Framework order – Limit”

These set of trace reveals a need for confrontation with the Client and this is mostly due to the fact that the same activity - *Change Approval for Purchase order* - is performed by two users on the basis of an apparent “four eyes” principle (*user 603* as a supervisor, *user 602* as an executive). Therefore, discovered model shows a loop that should be better *untangled* by identifying two different activities in place of one, for example: *Change Approval-Supervisor*, *Change Approval-Executive*. Process instances can be triggered by three start events, with different implications for the sequence of activities (Figure 12, Figure 13, Figure 14).



**Figure 12 – “2 way match” Process instance starts with *Vendor creates invoice* (basic sequence, no events looked at after “Clear invoice”)**

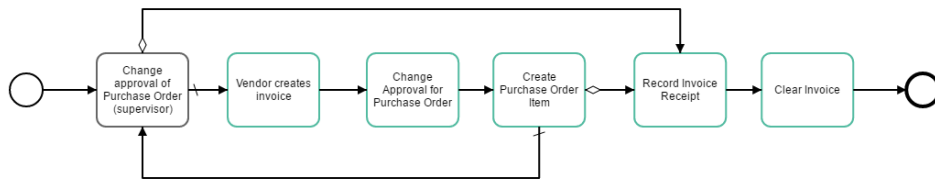


Figure 13 – “2 way match” Process instance starts with *Change Approval ... (by supervisor)*, basic sequence



Figure 14 – “2 way match” process instance starts with *Create Purchase Order*, basic sequence

The “four eyes” principle acts also in other phases of the sequence, usually after “Clear invoice” (namely, a sequence of “Change Approval..”, with the last one always executed by the *supervisor*) as if it were an authorization step even after months from Clear Invoice; this can be explained with both Item Type and Document Type associated with these traces. A framework order can have one or more invoices, with intermediate activities that act as periodic “authorization steps” to the payment or update to the payment accounting process. It would help discussing with the Client, given that these traces are mostly related to Real Estate and Energy spend areas.

## 7 “3 way match – After”

### 7.1 E.C. Purchase Order, Service (and Standard)

In 3.3 *Dataflows, documents and items: dimensions for modeling* we have learnt that some activities belong only to peculiar Document Type, as it is the case of E.C. Purchase Order. There is a set of activities, executed by a *batch* type of user usually in conjunction with Create Purchase Order (this one executed by any *user\_X*). The “SRM...” subprocess in Figure 15 can be triggered by a *user\_X* that creates a Purchase order or by a (None) Vendor that creates an Invoice.

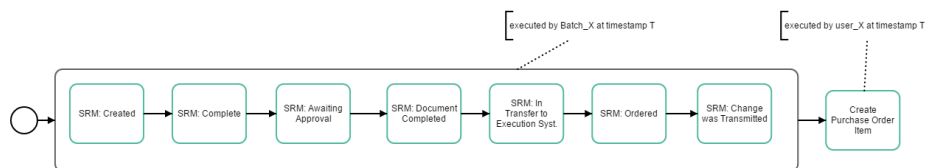


Figure 15 – “3 way After”, EC Purchase Order, Service (Part 1)

the “SRM...” sub process in Figure 16 instead occurs along the instance lifetime in two cases: after a “Vendor Creates Invoice” due to a probable mismatch of values between Invoice and Purchase Order, or after “Clear invoice” due to the fact that “Service” Item type can go through more than one invoice.

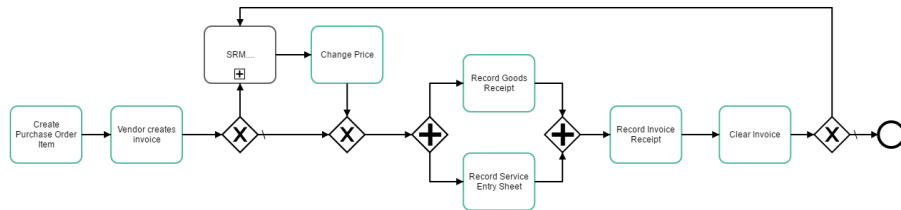


Figure 16 – “3 way After”, EC Purchase Order, Service (Part 2)

There is another difference about “SRM...” subprocesses in Part 1 and Part 2 and it is the executor. In Part 1, the subprocess is always executed by a *batch\_X*, while in Part 2, the first steps of subprocess (from “SRM: Created” to “SRM: Document Completed”) are executed by a *user\_X* and then by a *batch\_X*. It would be interesting to discuss whether a business rule could be implemented in order to automate the triggering of the subprocess.

Note that the sub-process can have more activities (the depicted ones are just the most frequent) and there is a case for “SRM: in transfer to execution system” that occurs in a small subset of traces after the Clear Invoice activity. What is important to focus in on is the time gap between these activities in such a small subset; in fact whenever the “Clear invoice” happened in 2018, then, at the beginning of 2019 the “SRM: in transfer...” occurs, probably because of a forced update of the system (if not for the event-log preparation), but hardly because of any planned process design.

The Item type is a Service one and this implies the presence of “Record Service entry sheet”. It is interesting to point out that this activity enters into the process instance in parallel with Record Goods Receipt (same timestamp). The fact here is that “Record Service entry sheet” is loaded by any vendor and exists *in the system* since, associated with a “None” user (the one for vendors), but it is later loaded *into the process instance* concurrently with the activity of Record Goods receipt which is executed by any *user\_X*. This shows a process design where an event pulls another event and can only be performed if the “pulled” one already exists.

Apart from the “Record Service entry sheet”, which won’t belong to it, traces within the category “3 way After, EC Purchase Order, Standard” show a similar process behavior as the above one.

## 7.2 Framework Order/Service and Standard P.O./Service

These traces differ from the E.C. Purchase Order/Service for the absence of “SRM...” activities and sub-process but show a similar pattern for their distinctive events (concurrency of Goods receipt and Service Entry sheet, users etc).

## 7.3 Standard P.O./Standard (Sub-contracting and Third Party)

Most of these traces show a process model as in Figure 11, in which activities “Record Goods receipt” and “Vendor creates invoice” are just parallel (not concurrent). As discussed about compliance, there is the case of “Remove Payment Block” occurring within the process, but this should not be interpreted as for the case of “3 way match, Before” traces. “Standard” traces also show the mentioned process change at the start of process instances (*3.5 Process drift and forced execution*).

## 8 “3 way match, Before”

### 8.1 E.C. Purchase Order/Standard

Traces with Document Type *EC Purchase Order* follow a process model with the above mentioned “SRM...” subprocess, which can occur at the start of the instance or within it in case of a change in price. In Figure 17 and Figure 18, Part 2 of the process is shown for traces where Goods are received before any Invoice and those where Invoice is received before any Goods.

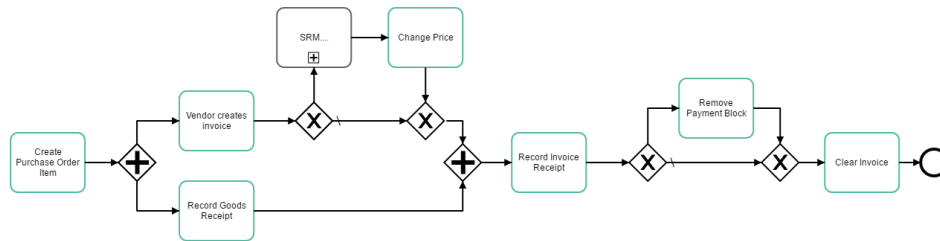


Figure 17 – “3 way Before, EC Purchase Order/Standard (Goods received before Invoice), Part 2

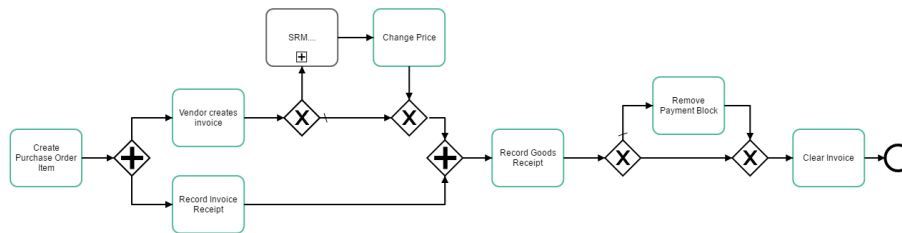


Figure 18 - “3 way Before, EC Purchase Order/Standard (Invoice received before Goods), Part 2

Part 1 of the process is actually the “SRM...” subprocess (model is simplified and doesn’t comprises cases with different Start events). Notes about “SRM...” subprocess (users involved etc) - already mentioned for “3 way After...” traces - apply to “3 way Before...” traces.

### 8.2 Standard P.O./Standard (Sub-contracting and Third Party)

These traces, the most common in the dataset, abide with the model already seen. The first part shows the change in process (see 3.5 Process drift and forced execution).

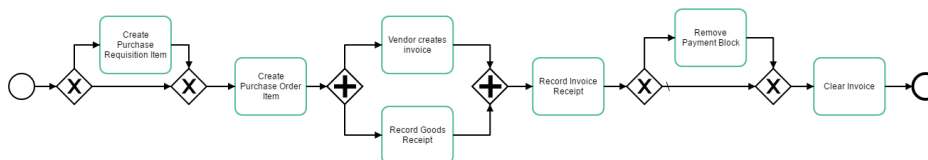


Figure 19 - “3 way Before, Standard P.O. (Goods received before Invoice)

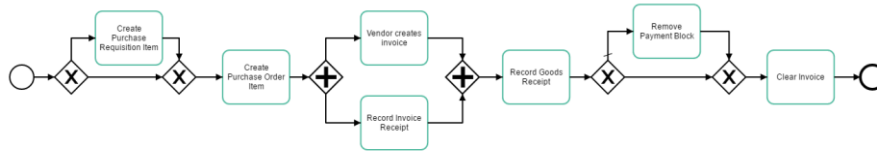


Figure 20 - “3 way Before, EC Purchase Order/Standard (Invoice received before Goods)”

It is worth mentioning a peculiar timestamp evidence involving two events (“Record Invoice receipt” and “Vendor creates invoice”). There are cases in which “Record Invoice receipt” occurs just hours before “Vendor creates Invoice” (executed in batches at 23:59). This is not a violation of model-logic, it is instead another example of the fact that events already existing *in the system* are later pulled *into the process* (see last part of 7.1).

## 9 Conclusions

Process mining can support decision making and enrich those data-related set of performance KPIs any Company would collect and represent in dashboards. This contribute has focused on the added information a process perspective can give to management. A critical approach to data has led to questioning hypotheses and assumptions, discussing about process drift and consequences of process change (forced execution) as well as making assumptions on *four eyes principle* and *segregation of duties* in some instances. Both organization and training issues have been mentioned, while a first step toward process discovery and modeling has raised even more questions than those originally asked for.

Process mining is about information, not IT tools; for such a reason, even if many different tools have been used, none is explicitly mentioned in the text. Besides, many information was originally derived with the aid of data querying languages and data visualization tools and then specifically investigated with mining algorithms.