Third International Business Process Intelligence Challenge (BPIC'13): Volvo IT Belgium VINST

Seppe K.L.M. vanden Broucke[1], Jan Vanthienen[1] and Bart Baesens[12]

[1] Department of Decision Sciences and Information Management, KU Leuven
Naamsestraat 69, B-3000 Leuven, Belgium
[2] School of Management, University of Southampton
Highfield Southampton, SO17 1BJ, United Kingdom
seppe.vandenbroucke@kuleuven.be

Abstract. For the Third International Business Process Intelligence Challenge (BPIC'13), a collection of real-life event logs from Volvo IT Belgium is analyzed. The provided logs contain events from an incident and problem handling system which is called VINST. This report presents results related to the following investigations. First, an open-minded exploratory analysis of the given event logs and second, answering the four specific questions posed by the process owner. To do so, we utilize both already existing as well as dedicated developed tools, and heavily combine traditional data analysis tools and process-oriented techniques; we indicate the existence of a gap between these two categories of tools and as such emphasize the importance of a hybrid approach in a process intelligence context throughout the report.

Key words: process mining, process analysis, data analysis, process intelligence, challenge

1 Introduction

1.1 Overview

This report presents the results we have uncovered related to the analysis of the Volvo IT data set in the context of the Third International Business Process Intelligence Challenge (BPIC'13). The data set consists of three real-life event logs from an incident and problem handling system called VINST.

The report is structured as follows. First, a high level overview of the data set and the VINST system is provided in Section 2, to familiarize readers with the provided material. Next, in Section 3, we perform an open-minded exploratory analysis using ProM[1] [1] and Disco[2] in order to get an initial idea about the event

[1] ProM is an extensible process mining framework for the discovery and analysis of process models and event logs. See http://www.processmining.org for more information.

[2] Disco is a commercial process discovery tool developed by Fluxicon and provides an easy-to-use interface together with extensive filtering capabilities. See http://fluxicon.com/disco/ for more information.

logs' composition and the VINST processes before moving on to answering the questions posed by the process owner in Section 4. The report is concluded in Section 5.

1.2 Methodology

We will make heavy use of both traditional data analysis tools and process-oriented techniques. We believe there to be a gap between these two categories of tools, causing difficulties when filtering, exploring and analyzing event based data sets. Reports submitted to previous BPI challenges, for instance, illustrate that practitioners and academics alike frequently perform initial filtering steps using spreadsheet or OLAP-based tools. Oftentimes, the same tools are used to derive descriptive statistics, before moving on to a more process-oriented view (e.g. process discovery or performance analysis), which these traditional data analysis tools lack. Since event logs are often handled as a large "flat table" (i.e. a listing of events), it is difficult to derive statistical information at the trace level (i.e. a grouping of events) rather than at the event level. Filtering out events based on complex control-flow based criteria is even more challenging. Consider for example the user trying to filter out all events belonging to a trace where two activities occur in parallel.

Vice versa, tools which are specifically geared towards the analysis of process based data are often lacking in terms of filtering or descriptive statistics. Consider ProM, for example, which contains a plethora of plugins for the discovery of process models alone, but does not provide strong methods to filter or explore the data set at hand. In ProM 6, only two helpful filtering plugins are provided by default:

- "Filter Log using Simple Heuristics", which allows to discard events based on their classification and traces based on their start/ending/contained events;
- "Filter Log by Attributes", which is able to filter on traces containing events with a specific attribute-value pair. Advanced filtering options (e.g. "value greater than x") are not provided.

The same can be said for plugins providing descriptive statics. Information about events per case, number of cases and events and other simple counts are easily retrieved, but no robust options are provided to derive other, more complex event data.

Due to the mismatch described above, practitioners and researchers currently often find themselves using an abundance of tools to perform their analysis tasks. While this is not an insurmountable problem, of course, the fact remains that available tooling towards the analysis of event logs can be improved. In fact, this is one of the challenges also listed in the process mining manifesto [2]. Disco stands out as a great contender, however, which has enabled practitioners and researcher to quickly filter and inspect event logs, but – as we will indicate later on – is not completely up to par with traditional data analysis tools when it comes to inspecting particular attribute distributions or correlations. Therefore, we will apply a hybrid approach in this report, switching back-and-forth between

(mainly) R and Disco to perform the desired tasks and combine statistical data analysis with process mining insights.

As a summary, Table 1 provides an overview of the differences between traditional data analysis tools and process oriented techniques. The main differences arise due to a different starting viewpoint on the data at hand. Traditional tools work with flat data tables containing record lines, whereas process oriented techniques are aware of the hierarchical nature between cases and events.

Other than the plethora of tools already available for (statistical) data analysis and process mining, it should be mentioned that some work has also been undertaken in order to add some process awareness – or, more precisely, sequence awareness – to data analysis tools. TraMineR, for instance, is an R package [3] for mining, describing and visualizing sequences of states, but contains support for event (i.e. transition) based data sets as well, such as found in business process event logs. The package offers a robust toolkit, but requires a relatively large amount of user expertise to use correctly. Furthermore, filtering data sets is somewhat difficult and the provided visualization options do not aim to discover a summarized process map from the given sequences. Many authors have also emphasized the need for sequential data support in traditional data base management systems [4–6]. One particular example of interest is SQL-TS [7–9], an extension for SQL which provides a simple extension for SQL to work with sequential patterns, using proposed "CLUSTER BY" and "SEQUENCY BY" clauses. This would provide a good starting point towards analyzing event based data sets. I.e. the "CLUSTER BY" clause indicates the field specifying the case

Table 1: Overview of the differences between traditional data analysis tool and process oriented techniques.

Characteristic	**Traditional Data Analysis**	**Process Oriented Analysis**
Example products	Excel, R, SAS, other statistical packages	Disco, ProM, other process mining tools
Data structure	Flat (table of records)	Hierarchical (cases with events)
Central object	Events	Cases
Process discovery (extracting visual process maps)	Hard	Easy
Statistical analysis (extracting distributions, correlations, relations)	Easy	Hard
Filtering	Easy at event level, hard at case level	Easy for some filters/tools (e.g. Disco), harder for others
Querying	Easy (e.g. SQL) at event level	Hard

identifier (e.g. "SR Number"), while the "SEQUENCE BY" clause indicates the ordering of events. Sadly, two drawbacks exists which makes this approach not yet fully suited for analyzing business processes. First, higher level structural behavior, such as loops and parallelism, is not "discovered" and can thus not be taken into account when constructing queries, or must be specified manually by the user (i.e. explicitly take into account all parallel orderings possible). Second, the language is not available for public use. Finally, we are also reminded of regular expressions [10] (or regular languages) and other expressive grammar languages such as linear temporal logic [11] (LTL) which are converted to automatons to support the querying of sequence based data. Again, these tools offer theoretically sound solutions to search sequence based data sets, but also require some more advanced user expertise and or not able to deal with higher level control-flow abstractions, so that they are quite far distanced from the easy-to-use traditional and common tools offer.

2 Data Set Overview

The BPIC'13 data set is obtained from Volvo IT Belgium and contains events pertaining to an internally developed Siebel-based incident and problem management system called VINST. Three log files are provided: one for incidents, one for open problems and one for closed problems[3].

The *incident management* process aims to restore normal service operations after the occurrence of a specific incident within SLA defined boundaries. Incident cases are first handled by a "first line" desk (the service desk and help desks) and escalated to second line and third line teams when the first line workers are not able to resolve the incident. Incident cases are assigned a priority level, which is calculated based on the impact (major, high, medium or low) and urgency (high, medium, low) of the issue, see Table 2.

Table 2: Priority levels for incident cases are calculated based on the impact and urgency of the incident. When incidents are not resolved within a specified time frame, urgency is increased automatically. Incidents cannot automatically migrate from one impact level to another.

	Major Impact	**High Impact**	**Medium Impact**	**Low Impact**
High Urgency	– (1)	4	7	10
Medium Urgency	– (2)	5	8	11
Low Urgency	– (3)	6	9	12

[3] DOI identifiers for the data sets are: doi:10.4121/500573e6-accc-4b0c-9576-aa5468b10cee, doi:10.4121/3537c19d-6c64-4b1d-815d-915ab0e479da and doi:10.4121/c2c3b154-ab26-4b31-a0e8-8f2350ddac11.

The *problem management* process tries to uncover the root causes behind incidents and implement fixes to prevent the occurrence of further incidents in IT-services operated by Volvo IT and includes activities to update internal knowledge bases with discovered findings. This process is primarily handled by second and third line teams. Contrary to the incident management process, there is no "push to front" system implemented for the problem management process which escalates cases among different service lines. The problem management and incident management processes thus work in parallel, where incidents are resolved as quickly as possibly in a "reactive" manner while the underlying root cause is fixed by a problem management case. These two workflows are supported by the Volvo IT "VINST" tool. The tool also supports other workflow processes such as *handle questions* and *major incident procedure*, but these are not incorporated in the BPI challenge data set.

The data set lists the following attributes for each logged event line:

- "SR Number" (or "Problem Number" for the problem management process): the service request case identifier.
  Example values: 1-364285768, 1-109135791, 1-147898401.
- "Change Date+Time": the time stamp of the logged event line.
  Example values: 2011-08-29T07:12:35+01:00, 2011-08-29T07:13:48+01:00.
- "Status" and "Sub Status": the current status of the case as changed by the logged event line.
  Example values: Queued/Awaiting Assignment, Accepted/In Progress, Accepted/Assigned.
- "Impact": the impact of the case.
  Example values: Medium, Low, High.
- "Product": the product involved in the case.
  Example values: PROD821, PROD236, PROD793.
- "Involved ST": the support team trying to solve the problem.
  Example values: V5 3rd, V30, V13 2nd 3rd.
- "Involved ST Functional Division": the support team's functional division.
  Example values: V3_2, C_6, E_10.
- "Involved Organization": the involved organisation line.
  Example values: Org line A2, Org line C, Org line V7n.
- "Organization Country": the location that takes the ownership of the support team.
  Example values: fr, se, nl.
- "Owner Country" and "Owner First Name": the person in the support team working on the case.
  Example values: France/Frederic, Sweden/Adam, Belgium/Bert.

Next section describes an exploratory analysis of the three event logs.

3 Exploratory Analysis

3.1 Descriptive Statistics

Using ProM and Disco, we perform some initial analysis on the given three event logs. Some rudimentary, process based descriptive statistics are listed in Table 3. We use the default classifier as stored in the event logs to classify events to a set of activities, namely the combination of "Status" and "Sub Status" attribute fields. For each event log, the total number of cases and events is listed, together with the number of distinct cases (the different variants). Next, the total number of event classes found in the event log, together with how many of these appear as the first or last activity in the traces contained in the event log is shown. Finally, Table 3 also lists the minimum, average and maximum case length (i.e. number of events).

Table 3: Process based descriptive statistics for the three event logs. For the three logs, the default event classifier ("Status" plus "Sub Status") is left as is.

	Event Log		
	incidents	**open problems**	**closed problems**
Nr. of Cases	7554	819	1487
Nr. of Events	65533	2351	6660
Nr. of Distinct Cases	2278	182	327
Total/Starting/Ending Nr. of Event Classes	13/9/8	5/5/5	7/6/1
Min./Avg./Max. Case Length	1/8/123	1/2/22	1/4/35

Especially for the "incidents" event log, the numbers indicate a large amount of variability. As shown by Fig. 1, a large number of trace variants exist which occur only once in the event log, leading to a skewed distribution in the number of cases for each variant. Next, table 4 shows an overview of the different attributes (described above) with their number of values and mean of frequencies of the values for the three event logs.

Let us now take a look at some performance related characteristics for the given event logs. When we plot the mean duration for the cases represented by a variant against the number of cases itself, a relation can be observed between the rarity of a trace variant and its running time, as depicted by Fig. 2a. One is free to remove these peculiar traces from the data set before continuing on with further analysis, we but choose to leave these exceptional cases alone for now. A similar observation can be made when plotting the case duration for all cases, for example as shown in Fig. 2b for the "incidents" log. Observe that a small number of cases exists which run for a high amount of time, with the longest trace taking 2.11 years.

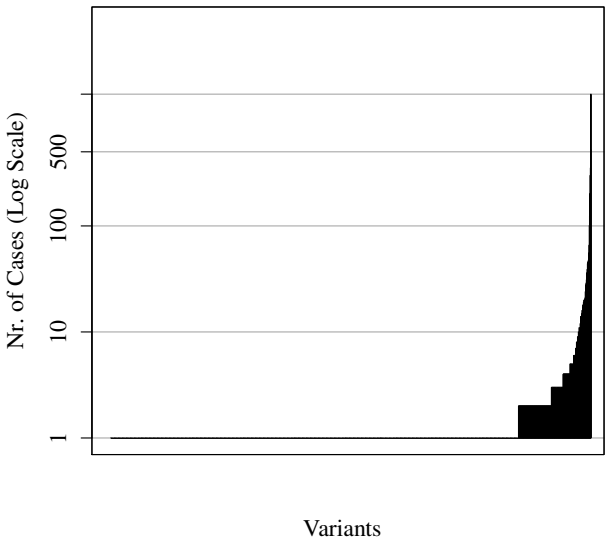
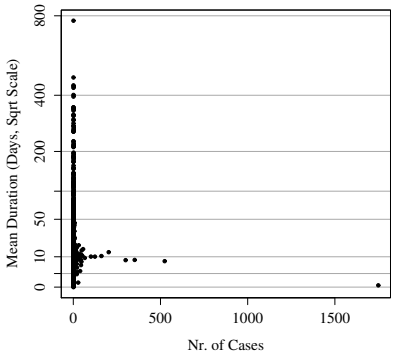


Fig. 1: Number of cases in the "incidents" log per variant. A "long tail" of 1934 variants exist which only occur once in the event log.

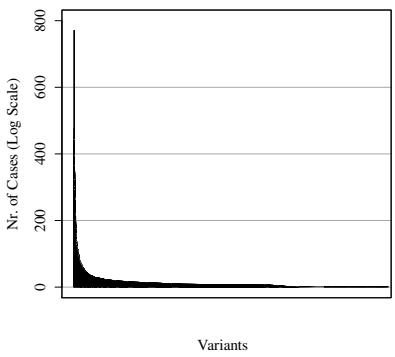
Next, Fig. 3 inspects the relationship between a case's duration and the involved product or impact level of the case. We observe that there exists a strong correlation between the product a case relates to and the required running time for that case. Indeed, this is to be expected, as is seems reasonable to

Table 4: Descriptive statistics for the attributes contained in the event logs. For each attribute, the absolute number of possible values is shown, together with the mean of the frequencies of the values (between parentheses).

Attribute	**Event Log**		
	incidents	**open problems**	**closed problems**
Status	4 (16383.25)	3 (783.67)	4 (1665.00)
Sub Status	13 (5041.00)	5 (470.20)	7 (951.43)
Impact	4 (16383.25)	4 (587.75)	4 (1665.00)
Product	704 (93.09)	139 (16.91)	337 (19.76)
Involved ST	649 (100.98)	1 (2351.00)	324 (20.56)
Involved ST Functional Division	24 (2730.54)	26 (90.42)	29 (229.66)
Involved Organization	25 (2621.32)	1 (2351.00)	15 (444)
Organization Country	23 (2849.26)	1 (2351.00)	17 (391.76)
Owner First Name	1440 (45.51)	240 (9.80)	585 (11.38)
Owner Country	32 (2047.91)	14 (167.93)	21 (317.14)



(a) Scatter plot showing the mean duration in days for each variant against its number of occurrences. Many exceptional variants also exhibit a long duration.



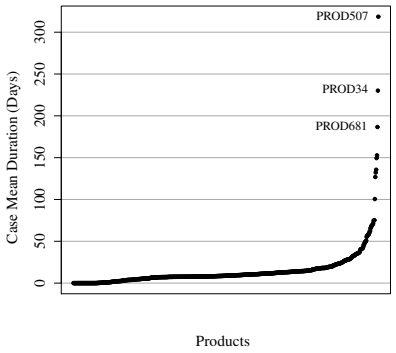
(b) Case durations (shown in days) for all cases included in the event log.

Fig. 2: Performance related characteristics for the "incidents" event log.

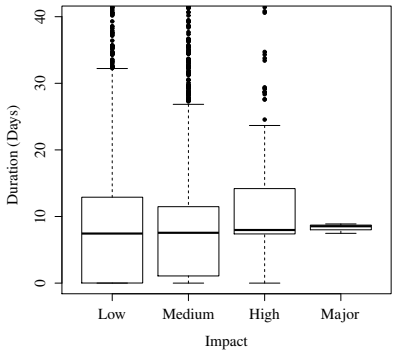
assume that some products (such as "PROD507", "PROD34" and "PROD681" highlighted in Fig. 3) lead to incidents which are harder to resolve, due for example to the complexity of the product. Concerning impact levels, the box plots shown in Fig. 3 do not immediately reveal a great difference in duration *medians*, although there does exist a significant difference between the *mean* durations over the impact levels (which are 9.7, 13.7. 16.2 and 8.3 days for "Low", "Medium", "High" and "Major" impact respectively).

We can also derive performance related statistics for the support team member ("Owner First Name") and countries ("Owner Country"). However, contrary to products, support teams and member are event-attributes, and can thus vary within a case. When we would group traces as done above (e.g. a trace belongs to the group associated to "Sam" when one of its events was executed by Sam), this would lead to a skewed distribution as perhaps only one event in a long trace was executed by a certain support team member. We are thus interested in knowing the duration of certain tasks executed by the various support team members, rather than the duration of the whole case.

Sadly, the provided event logs do not contain starting and stopping times for each activity. Instead, activities are logged as atomic events, i.e. only with a single time stamp, and are assumed to have a duration of zero (instant completion). Still, when relating task durations to support team members, we nevertheless would like to have some idea about how long the case was being handled by a certain person before moving on to the next activity (and, perhaps, some other support team member). Therefore, we apply a simple pre-processing step where we assign a completion time to each event equal to the (starting) time stamp



(a) Mean case duration grouped per product. Observe that there exists a strong correlation between the products involved in a case and the required running time for that case.



(b) Box plots depicting case duration statistics per impact level. Although no strong difference among the medians (shown by the box plots) becomes apparent, the variance decreases for higher impact levels. The means of the different impact levels also significantly differ at the 95% confidence level.

Fig. 3: Case duration as depending on the product involved in the case, or the case's impact level ("incidents" event log).

of the event following directly after in the same trace. If there is no such event, we just set the completion time equal to the starting time and assume this final event was instantly completed. This allows us to derive how long a particular case "stayed" with a particular activity, and hence, support team member[4]. Once done, we can easily extract various sorts of information for the support team members and countries. Table 5 provides an overview of mean activity durations for activities, support team members and support team countries, as well as a task-originator matrix for the support team members.

As always, these initial exploratory results must be carefully interpreted to avoid reaching to erroneous conclusions. The common warning given when dealing with performance-originator oriented statistics indeed states that the fact that someone is taking longer to do their job might simply be because this worker commonly deals with harder jobs – which take longer to resolve. Additionally, using the mean duration may lead to misleading results under skewed distributions; for example, the median duration for Anne Claire's activities is a just a few seconds. The standard deviation for the workers with high mean

[4] Note that we assume the time stamp present in the logs to denote the starting time of a particular activity. Only when this activity is completed does the next event start.

Table 5: Performance related overview metrics for support team members and activities for the "incidents" event log.

(a) Mean activity durations for all activities based on derived completion times. Again, "Status" plus "Sub Status" is used as a classifier to construct activities.

Activity	Mean Duration (Hours)
Accepted/Wait - Vendor	5.48
Accepted/Wait - Implementation	5.40
Accepted/Wait - Customer	4.81
Completed/Resolved	4.69
Accepted/Wait - User	4.58
Accepted/Wait	4.08
Accepted/Assigned	1.51
Queued/Awaiting Assignment	1.09
Accepted/In Progress	0.44
Completed/In Call	0.22
Completed/Closed	0.15
Unmatched/Unmatched	0.01
Completed/Cancelled	0.00

(b) Mean activity durations for the support team members ("Owner First Name") based on derived completion times. Only the highest ten members are shown.

Owner First Name	Mean Duration (Days)
Anne Claire	103.44
Earl	60.16
Mica	48.91
Elaine	44.79
Ray	42.75
Cincellia	42.27
Bozidar	36.87
Lucas	27.47
Brice	23.29
Minkyu	22.93

(c) Mean activity durations for the support team countries ("Owner Country") based on derived starting times. Only the highest ten countries are shown.

Owner Country	Mean Duration (Days)
Denmark	19.26
Austria	14.76
Netherlands	12.46
Spain	4.65
Turkey	3.85
Germany	3.76
United Kingdom	3.11
France	2.94
Russian Federation	2.86
Korea	2.60

(d) Task-originator matrix for the support team members. Only the first ten team members are shown, sorted by the number of activities executed (summed last row).

	Siebel	Krzysztof	Pawel	Marcin	Marika	Michael	Fredrik	Piotr	Andreas	Brecht
Queued/Awaiting Assignment	538	206	125	93	90	117	104	92	53	91
Accepted/Assigned	23	189	31	73	193	10	31	68	0	44
Accepted/In Progress	12	616	479	332	297	320	267	304	286	222
Accepted/Wait	27	5	32	7	4	12	18	18	6	0
Accepted/Wait - Implementation	8	1	1	0	0	0	0	5	0	0
Accepted/Wait - User	173	56	123	71	12	32	69	33	63	75
Accepted/Wait - Vendor	7	0	0	5	0	0	10	0	33	0
Accepted/Wait - Customer	1	1	0	0	0	1	0	0	0	0
Completed/Cancelled	0	0	0	0	0	0	0	0	0	0
Completed/Resolved	0	87	88	80	9	56	83	16	72	38
Completed/In Call	0	5	24	10	0	31	3	16	29	7
Completed/Closed	5373	7	22	17	0	8	0	2	0	0
Unmatched/Unmatched	0	0	0	0	0	0	0	0	0	0
Sum	6162	1173	925	688	605	587	585	554	542	477

durations is high, caused by a few (or even just one) activities taking a long time to be passed forward. This is evidenced by Fig. 4, which shows a small number of activities with a very high duration (i.e. more than one year). The process owner can inspect these activities and the cases they belong to in order to verify if further action should be undertaken. Furthermore, keep in mind that we have derived activity durations based on a single time stamp present in the log files, which might differ from the actual work time.

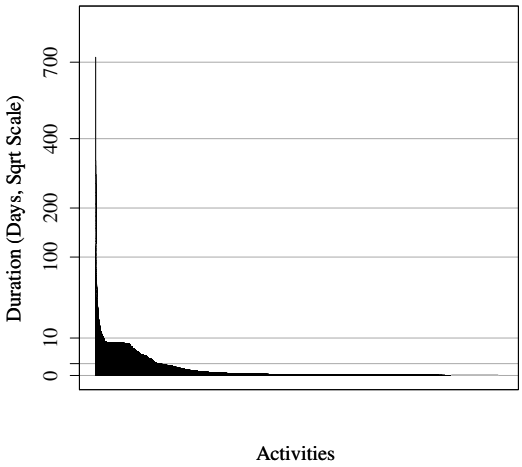
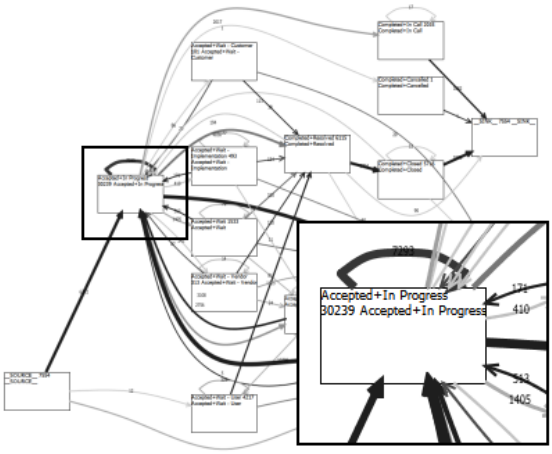


Fig. 4: Durations for all activities in the "incidents" event log. Notice the few number of events consuming an unrealistic amount of time.

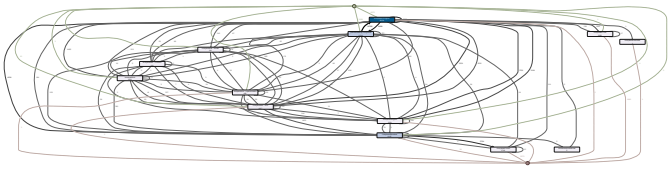
We limit the statistical description to the overview provided above, as not to consume too much space. Nevertheless, this section illustrates the power of using simple, "flat data" oriented tools already widely available to get a first, solid introspection in the performance and characteristics of running processes. However, since these tools lack a process oriented perspective, it is difficult to derive statistical information at the trace level (a grouping of events) rather than at the event level, as was already encountered in this section. This already emphasizes the need for a hybrid approach. Therefore, we also briefly explore the data set using process specific tools, namely ProM and Disco.

3.2 Process Mining

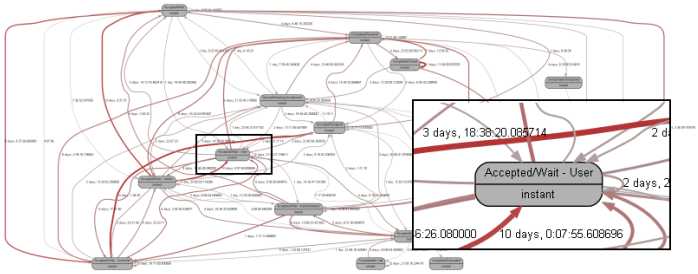
Mining the "incidents" event log using HeuristicsMiner [12] in ProM yields a result as depicted in Fig. 5, i.e. a "spaghetti" like model where all activities appear to be connected with one another. Using Disco with all activities and paths visible gives a similar (albeit prettier) outcome.



(a) Model mined with HeuristicsMiner in ProM (we use a custom created plugin which allows to provide the graph annotations).



(b) Model mined with Disco (all activities and paths shown).



(c) The same model mined with a custom built process discovery tool, which runs on top of Pandas (another statistical data analysis tool; see http://pandas.pydata.org) and extracts process flows together with summarizing metrics from a flat event table. This is then exported to a process graph (using Graphviz). This illustrates that the integration of process-oriented perspectives within traditional statistical tools is indeed possible.

Fig. 5: Mining the "incidents" log as is yields a so called hard to read "spaghetti" model, typical for non-structured ticketing and issue handling processes.

Let us see whether we can derive a "straight-through" process, a simple process map explaining the multitude of behavior found in the event log. Filtering on the frequent variants provides a solid means to do so. Fig. 6 shows the process map derived from the five most frequent variants, accounting for 41% of all traces in the "incident" log.

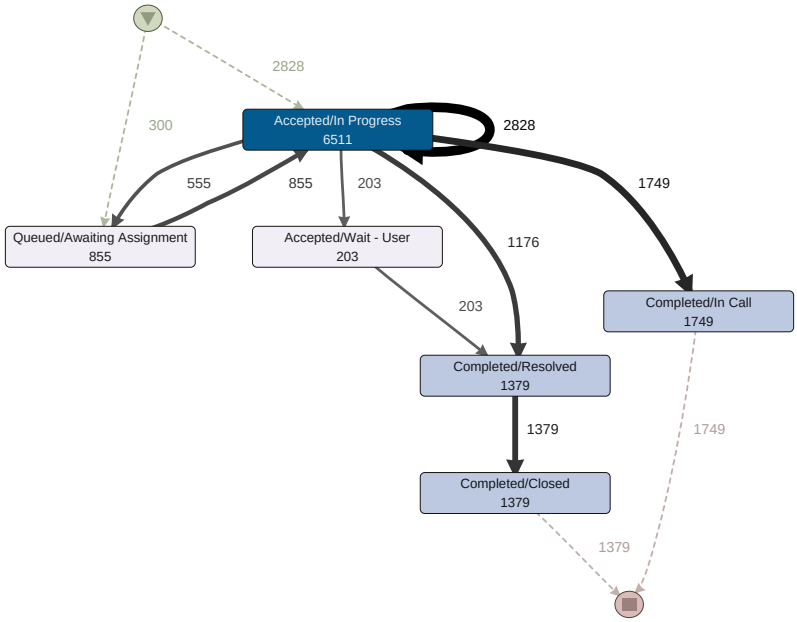


Fig. 6: Process map (all activities and paths shown) derived from the five most frequent variants in the "incidents" log.

When we just use the "Status" attribute as a classifier, the resulting process model becomes even simpler, and just shows all transitions between the "Accepted", "Queued", "Completed" and "Unmatched" statuses, see Fig. 7.

Since the process described a ticket based issue handling system, and since the event logs do not contain start and ending times by itself (unless approximated with a pre-processing step as done above), we can assume that the process does not contain any parallel behavior, i.e. process case is passed on from one activity to the next, without an activity create two or more concurrent flows of work. That said, it is thus possible to view the process as a Markov chain, i.e. as shown in Fig. 8 (once more using both "Status" and "Sub Status" fields), similar

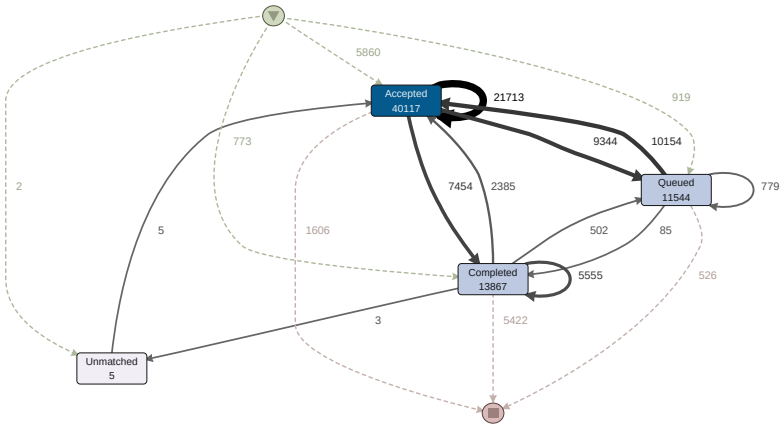


Fig. 7: Process map showing all transitions between the statuses for the "incidents" log.

to the process maps obtained by Disco, the depicted graph is also able to quickly highlight the most important flows from one activity to the next.

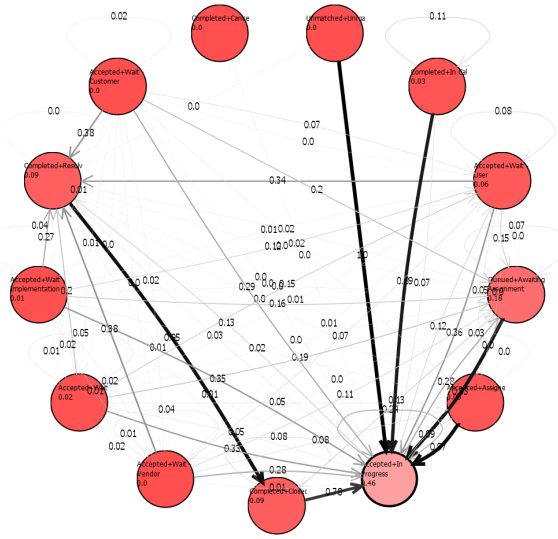
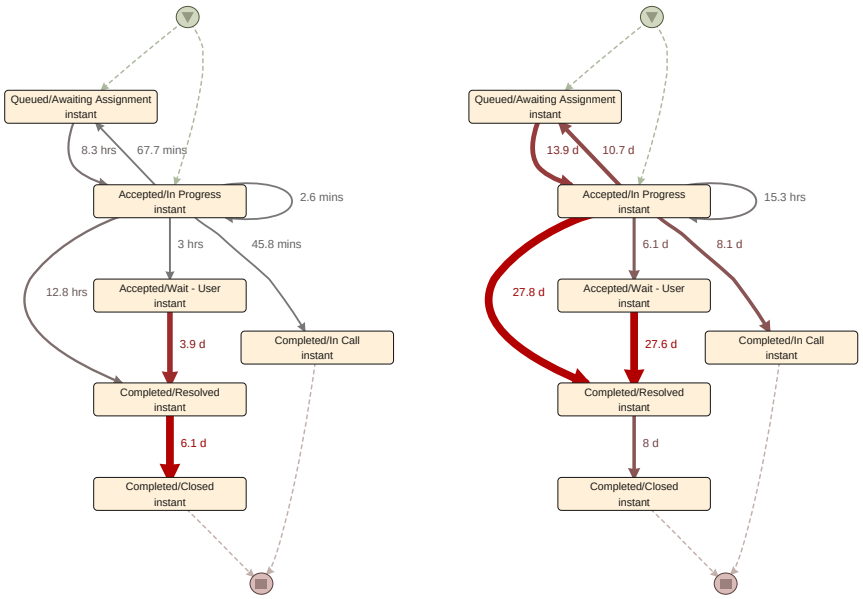


Fig. 8: Markov chain for the "incident" log (we use a custom created ProM plugin to extract the Markov chain).

We can also easily show some initial performance focused process maps. Fig. 9 depicts the maximum and mean waiting times between activity occurrences. Note the discrepancy between the mean duration and the maximum duration, also indicating the presence of low-frequent exceptional cases which run far longer.



(a) Mean durations. (b) Maximum durations.

Fig. 9: Two process maps showing performance related metrics. Note the discrepancy between the mean duration between activity flows and the maximum duration. The mean durations show the bottlenecks for average cases, whereas the maximum duration highlights a different part of the process model which acts a bottleneck, albeit only in some exceptional cases.

One can continue to apply various other filters and views in order to extract additional process maps. Fig. 10, for instance, shows the process map obtained for the "incidents" log after filtering on performance (keep only quickest 95% of cases, i.e. taking less than about one month), endpoints (cases should start with either a "Accepted" or "Queued" activity and end with "Completed), time frame (first of January 2012 onwards), and variation (use only cases whose sequence of activities is shared by at least two cases). The process map is quite readable and easy to interpret, but does not help further towards gaining more insights,

which requires first stating the goals and questions one wants to investigate. We will do so in the following section.

4 Answering the Process Owner's Questions

This section explores the questions posed by the process owner in the context of the BPI challenge. Four broad questions were outlined, which will be dealt with accordingly in the following subsections.

4.1 Question 1: Push to Front

This question only applies to the "incidents" event log. In this process, a strategy is put in place which states that most of the incidents should be resolved by the first line support teams. The process owner is interested in knowing how many (and which) cases escalate to a second or third team.

The provided case description is somewhat ambiguous concerning which criteria exactly determine whether an event was executed by a first line team. It is stated that "The second line could be a support team within Org line C or a support team working with Org line A2." This might lead one to believe that all teams not belonging to either "Org line C" or "Org line A2" are the first line teams. However, inspecting the "Involved ST" field in the data set includes names such as "V5 3rd" and "V13 2nd 3rd", which leads us to believe that the level assignment is included in this moniker. Therefore, we will add an extra field to the data set. Support team names including the strings "2nd" or "3rd" will be regarded as an above first level team (second, third or both). Names not including such string shall be regarded as first level teams.

Next, we import the data set in Disco, but specify the "Involved ST" as the activity and derive the following case counts:

- 7554 cases in total, creating an unstructured model.
- 4782 cases exist (63%) which did not require intervention from a second or third line support team. This is obtained by applying an attribute filter.
- 2772 cases (the others) did pass through a higher level team.
- 704 of these cases even immediately start in a higher level.
- 556 cases start and stay within higher level teams during their complete life cycle. This is a peculiar result. Fig. 11 shows the three most involved support teams (all paths shown) where this happens.

"For what products is the push to front mechanism most used?" We export the cases found with Disco and turn to R to answer the posed subquestions. After some straightforward operations, we derive for each product the number of times the product got involved in a case passing through a higher level support team. Fig. 12 shows a scatter plot showing the number of cases each product was involved in for which the case did not go through second or third level support teams (blue color) or did escalate to second or third level

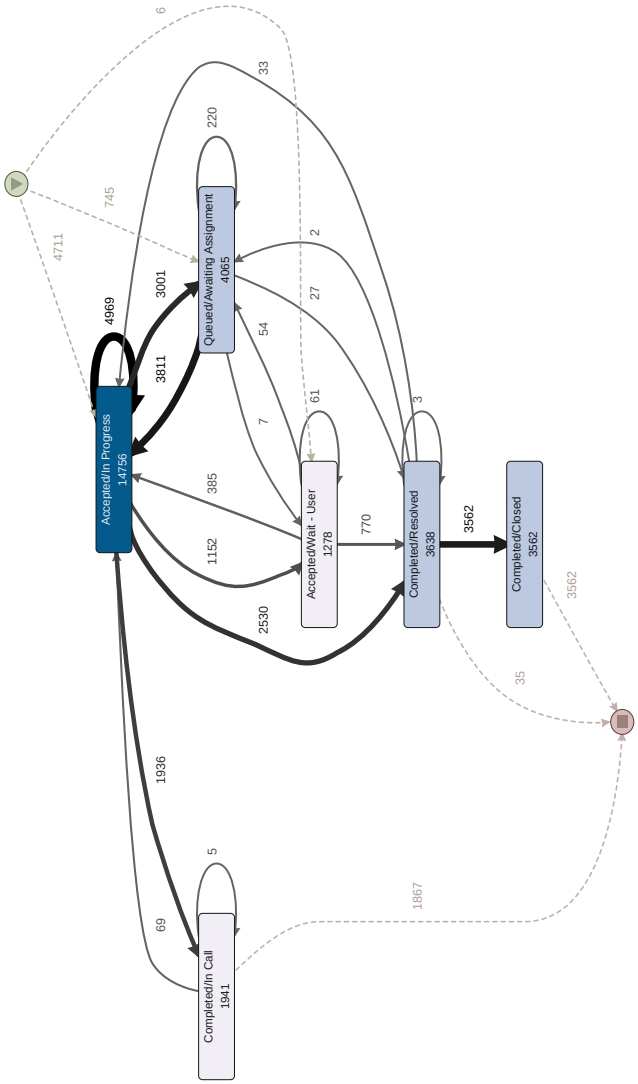


Fig. 10: Process map for the "incidents" log after filtering on endpoints, time frame, performance and variants. Activities with an absolute frequency less than a thousand are filtered out as well. All paths are retained.

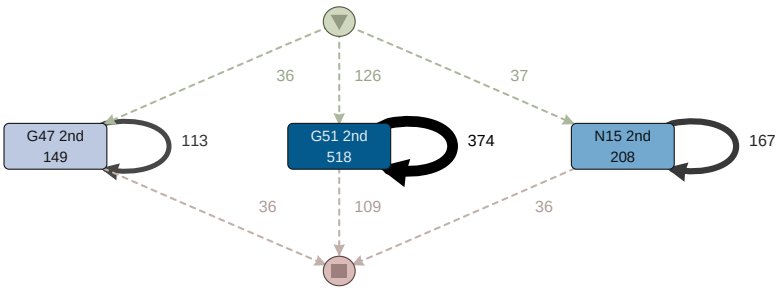


Fig. 11: 556 cases stay within second or third level support teams during their complete life cycle.

support teams (red color). We can indeed notice some products being escalated frequently to higher levels. Table 6 shows the twenty products with the highest number of cases with second or third level support team involvement, together with the number of cases without escalation.

"What is the impact of incident escalation on performance?" We did not find a clear relationship between mean duration for all cases *relating to a certain product* and the *number of times* the product was involved in a case escalated to second or third level support teams (or not involved). However, there is a significant difference in means of case durations for products which were *never* involved in an escalated case and product which were (8.6 versus 16.5 days). There also exists a significant difference in means of case durations for *cases* pushed to second/third level teams and cases that did not, i.e.: an average case duration of 20.5 days versus 7.2 days. Avoiding higher level escalation is thus indeed an important goal to be maintained.

We have also inspected the impact of escalated incident cases on the duration of problem handling cases, as the case states that cases ending up in second or third line support teams "disturbs them from doing their core activity (which is generally not support)" and that handling problem cases are "primarily the responsibility of 2nd and 3rd line support teams." It might thus be interesting to see if we can find a correlation between the performance of second and third line teams in the problem handling process (one of their main tasks) and the number of active escalated incident cases (which disturb them). Fig. 13 shows the resulting outcome. The thick lines represent the number of cases over time (all incident cases, escalated incident cases and problem cases – both open and closes), whereas the dotted lines show the mean and median duration of problem cases active in the same time window. No significant correlation can be derived. This might be due to the data set (incident cases arrive neatly over time but

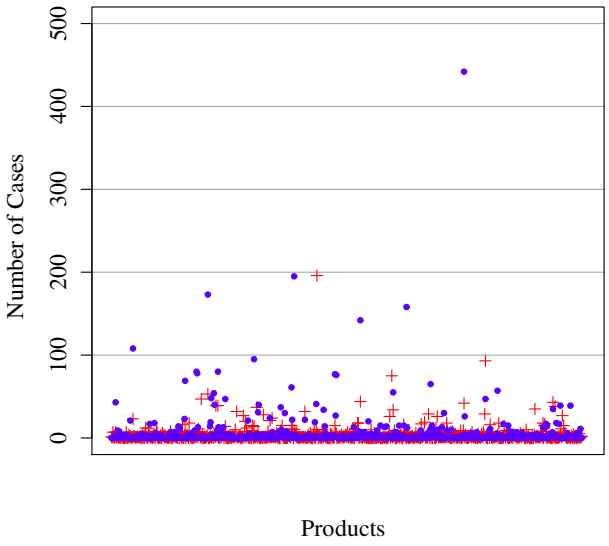


Fig. 12: Scatter plot over the products showing the number of cases the product was involved in for which the case (i) did not go through second or third level support teams (blue color – points) or (ii) did escalate to second or third level support teams (red color – crosses).

Table 6: Ten products with highest number of cases with second or third level support team involvement.

Product	**Number of Cases with Second or Third Level Involvement**	**Number of Cases with First Level Involvement Only**
PROD424	196	686
PROD698	93	47
PROD542	75	0
PROD253	53	173
PROD243	47	2
PROD494	44	142
PROD802	43	35
PROD660	42	442
PROD267	39	80
PROD264	39	1

almost always keep running until the middle of 2012). Also, in our analysis, cases are seen as "escalated" even when they are not yet escalated at a specific point in time (note however that this does not impact the median duration of problem cases, which stays stable over time anyway). More data is needed to perform a more rigorous analysis, or it might simply be that there exists no correlation between these two specific properties.

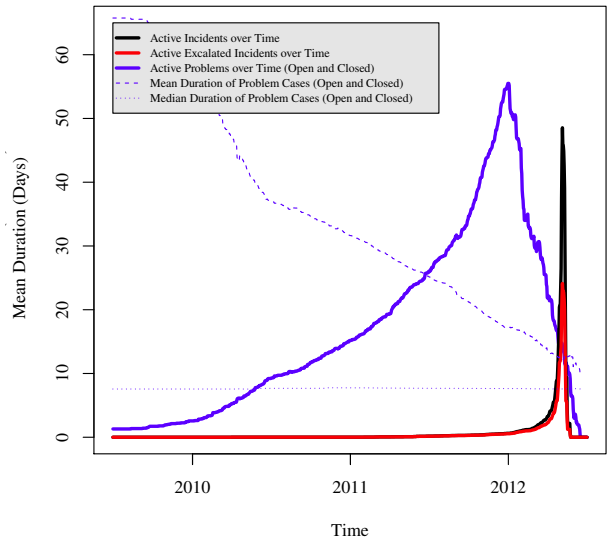


Fig. 13: Impact on number of escalated incident cases on duration of problem cases over time.

"Where in the organization is the push to front process most implemented, specifically if we compare the Org line A2 with the Org line C? What functions are most in line with the push to front process?" Finding out which organizational lines (or support teams or functional divisions) were best able to deal with incidents without escalating them is non-trivial, as these are event-related attributes and do not necessarily remain constant over a complete case, so that deriving event based counts (e.g. with Disco) leads to skewed distributions when cases contain varying number of events.

For the organization lines A2 and C, we can apply an attribute filter to retrieve the number of cases. Of the 2772 cases which were escalated to higher level teams, 1298 of them were handled by "Org line A2" at some point, whereas 1950 cases were handled by "Org line C".

For the involved support teams (and functional divisions), it is bothersome to re-apply a filter for each different support team. Hence, we switch once more to R. We aggregate all escalated and non-escalated cases by support team and use the length of the unique list of Case IDs ("SR Number") as the aggregation function to get, for each support team, the number of cases the team was involved in (for at least one activity) (only taking cases into account which escalated or did not escalate to a higher level team). We then order the teams based on their relative frequency difference between escalated and non escalated cases to account for the different number of cases. Table 7 shows the resulting overviews. An overview for organization lines was included once more. Note the absence of "Org line C" is this listing: although this line handled 1950 cases which were escalated, the organization also handled 4176 cases which were not escalated, so that this organization is dropped after deriving the relative frequency difference.

4.2 Question 2: Ping Pong Behavior

The case description states: "It occurs that support teams start to send incidents to each other again and again (ping pong) which of course is an unwanted situation." The business owner is interested in finding out about such cases. First, we have to be clear about what exactly is meant with "ping pong" behavior:

- The attribute of interest is the involved support team;
- Ping pong is defined as a case starting with support team A, moving then to another support team B (ping) and finally coming back to A later on (pong). For example, we will consider the following support team trajectory as being acceptable: $\langle A, B, C, D\rangle$, although many support teams are involved, while the following is not: $\langle A, B, C, A, D\rangle$.
- No information about status requirements are given. Note that it is however possible to perform an additional check here, e.g. also require that the "Queued/Awaiting Assignment" status is revisited again when switching support teams.

Let us start with investigating the "incidents" log. We first filter out all cases which stay with the same "activity" (here: the involved support team), by using a "Follower" filter in Disco which allows any activity succession but requires different values for each matched pair. This leaves us with 50% of the "ping" cases (3820 cases). Next, we want to get the "pong" behavior (returning to a previous support team). This is less easy to retrieve with Disco, as the follower filter cannot be applied here. As such, we search for "tandem repeats" using R and the following aggregation. We aggregate by Case ID ("SR Number") and apply a custom function to signpost cases containing tandem repeats. The method we apply is a bit involved and further detailed in Listing 1. We find 1258 cases with ping pong behavior.

We can then export these cases to Disco again for visual validation and analysis. Fig. 14 shows a simplified process map containing the ping pong cases with some support teams hidden but all paths shown. Back-and-forth ping pong behavior is indeed clearly visible.

Table 7: Ten highest organization lines, support teams and functional divisions ordered based on relative difference between escalated and non escalated cases.

(a) Organization lines.

Organization Line	Nr. of Cases with Second or Third Level Involvement	(%)	Nr. of Cases with First Level Involvement Only	(%)
Org line A2	1298	46.83%	501	10.48%
Org line B	408	14.72%	219	4.58%
Other	318	11.47%	242	5.06%
Org line V11	85	3.07%	41	0.86%
Org line G1	50	1.80%	4	0.08%
Org line V2	37	1.33%	32	0.67%
Org line G2	23	0.83%	14	0.29%
Org line F	13	0.47%	4	0.08%
Org line E	15	0.54%	16	0.33%
Org line V3	5	0.18%	4	0.08%

(b) Involved support team.

Support Team	Nr. of Cases with Second or Third Level Involvement	(%)	Nr. of Cases with First Level Involvement Only	(%)
G97	486	17.53%	490	10.25%
G92	123	4.44%	45	0.94%
D1	58	2.09%	11	0.23%
D2	94	3.39%	78	1.63%
D4	106	3.82%	104	2.17%
V17 3rd	33	1.19%	6	0.13%
D5	86	3.10%	101	2.11%
G34 3rd	27	0.97%	2	0.04%
G49	29	1.05%	7	0.15%
D6	36	1.30%	28	0.59%

(c) Functional division.

Functional Division	Nr. of Cases with Second or Third Level Involvement	(%)	Nr. of Cases with First Level Involvement Only	(%)
(blank)	1008	36.36%	378	7.90%
E_10	687	24.78%	3	0.06%
A2_2	385	13.89%	71	1.48%
A2_4	302	10.89%	27	0.56%
A2_3	244	8.80%	18	0.38%
A2_1	515	18.58%	567	11.86%
E_6	154	5.56%	3	0.06%
E_5	328	11.83%	346	7.24%
E_1	77	2.78%	10	0.21%
A2_5	75	2.71%	41	0.86%

Listing 1: Finding ping pong behavior.

```
# Assume the following trace with ping pong behavior
# (return to b):
> trace <- c('a', 'a', 'b', 'c', 'b', 'd')
# "Match" all elements equaling b:
> match(c(trace), 'b', nomatch=0)
[1] 0 0 1 0 1 0
# Take the difference between elements:
> diff(match(c(trace), 'b', nomatch=0))
[1]  0  1 -1  1 -1
# 0: stays at 'b', -1: leaves from 'b', 1: enters 'b'
# We thus check if 'b' is 'entered' multiple times by finding
# elements equaling 1:
> match(diff(match(c(trace), 'b', nomatch=0)),
+       1, nomatch=0)
[1] 0 1 0 1 0
# And summing:
> sum(match(diff(match(c(trace), 'b', nomatch=0)), 1,
+   nomatch=0)) > 1
[1] TRUE
# However, it could be that our trace starts with the element
# we're finding a tandem repeat for, we thus create an
# artificial start event:
> sum(match(diff(match(c('__START__', trace), 'b', nomatch=0)
   ), 1, nomatch=0)) > 1
[1] TRUE
# We've only tested for 'b', but need to do
# so for each element:
> lapply(unique(trace),
+   function(el) {
+     sum(match(diff(match(c('__START__',trace), el,
+       nomatch=0)), 1, nomatch=0)) > 1
+   }
+ )
[[1]] [1] FALSE
[[2]] [1] TRUE
[[3]] [1] FALSE
[[4]] [1] FALSE
# Summing: is there an element which is tandem repeating?
> sum(unlist(lapply(unique(trace),
+   function(el) {
+     sum(match(diff(match(c('__START__',trace), el, nomatch
   =0)), 1, nomatch=0)) > 1
+   }
+ ))) > 0
> [1] TRUE
```

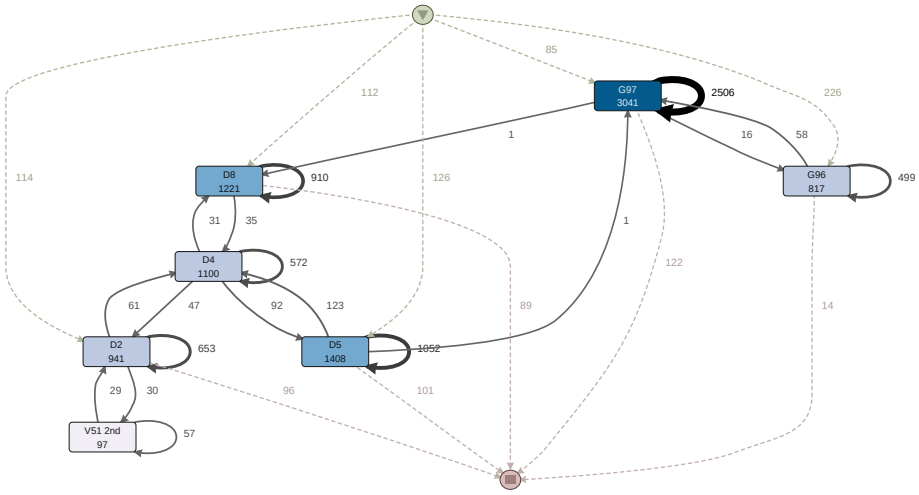


Fig. 14: Simplified process map (some support teams hidden) clearly showing back-and-forth ping pong behavior.

"Is there a correlation between the ping pong behavior and the total life time of an incident?" Fig. 15 shows a scatter plot with case durations separated by cases with (colored red) and without ping pong behavior (colored blue). Contrary to the statement made in the case description, no significant correlation becomes apparent. A Welch Two Sample t-test confirms that the means of 1.47 (ping pong) days and 1.35 days (no ping pong) are not significantly different at the 95% confidence level.

Table 8 summarizes the findings of ping pong behavior for the three event logs. Since the problem processes do not contain separate fields for the involved support team and support team functional division, we have only used the latter for these logs. We note that the relation between duration and ping pong behavior is not so clear as it may appear at first sight. For instance, we find that closed problems which do exhibit ping pong behavior are closed faster than those which do not. One might suggest to inspect the correlation between case duration and the number of ping pongs performed, but we did not find any particular pattern when doing so for the "incidents" log (using the number of support teams which ping ponged out of the total support teams involved in the case).

"What are the functions, organizations, support teams responsible for most of the ping pong? What products are most affected by it?" Due to space limitations, we only include the results for the "incidents" log. The same approach can be applied to the other logs. Again, we aggregate at the trace level, rather than the event level to account for skewed results. Table 9 lists the ten highest products, support teams, functional divisions and organization lines for the cases containing ping pong behavior.

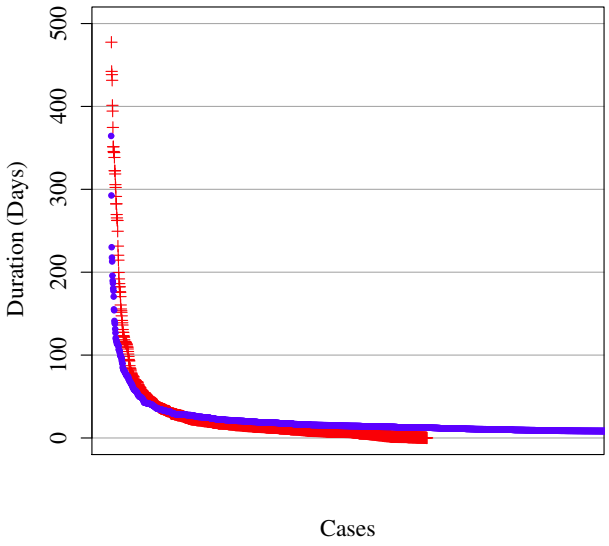


Fig. 15: Scatter plot showing the durations for cases with ping pong behavior (red – crosses) and without (blue – points) for the "incident" event log.

Table 8: Summarized findings of ping pong behavior for the three event logs. "Ping" behavior denotes cases which jump from one support team (or functional division) to another. "Ping pong" behavior are cases which also return to an earlier visited team or division.

Event Log	Nr. of Cases	Nr. of Cases with Ping Behavior	Nr. of Cases with Ping Pong Behavior	Significant Mean Duration Difference
incidents (involved support team)	7554	3820	1258	No (p-value: 0.10)
incidents (support team functional division)	7554	2574	1159	No (p-value: 0.31)
open problems (support team functional division)	819	71	21	Yes (p-value: 0.00), 22 versus 3 days
closed problems (support team functional division)	1487	210	92	Yes – inverse (p-value: 0.00), 22 versus 43 days)

Table 9: Products and teams exhibiting the most ping pong behavior (shown as number of cases). Only the ten highest items are shown.

Product (Nr. of Cases)		Support Team		Functional Division		Involved Organization Line	
PROD424	126	G97	286	V3_2	606	Org line C	972
PROD542	74	G96	242	A2_1	567	Org line A2	617
PROD236	41	3D4	192	(blank)	470	Org line B	206
PROD455	37	D5	126	E_10	247	Other	115
PROD776	31	D8	121	A2_2	149	Org line V11	109
PROD258	29	D2	115	E_5	122	Org line V7n	76
PROD697	29	G230 2nd	78	A2_3	82	Org line G4	38
PROD305	27	V37 2nd	76	E_6	71	Org line G1	34
PROD789	25	G179	71	A2_5	63	Org line E	30
PROD215	24	V32 2nd	64	A2_4	59	Org line V7	17

4.3 Question 3: Wait User Abuse

The challenge case states: "People try to find workarounds that stop the clock from ticking. One way of doing this is manually giving an incident the sub status 'wait user'. Although there are guidelines not to use this sub status, some people are breaking this guideline.

We again use the "incidents" log to inspect this behavior. We're interested in the "Sub Status" "Wait - User", which only occurs together with the "Accepted" main status. We can thus filter on this activity which results in 33% of the cases containing the "Accepted/Wait - User" activity.

Who is making most use of this sub status (action owner)? What is the behavior per support team, function, organization etc? To delve deeper into this behavior, we inspect how long cases stay within the "Accepted/Wait - User" state before another state becomes active, and will correlate this with other properties. Table 11 provides an overview of the ten support team members having spent most time in the "Wait - User" state, ranked by the median to account for exceptional outliers. Again, one should take care not to derive conclusions too quickly. We could, for example, filter out owners such as "Susanne" whose minimum duration does not exceed a certain threshold. Alternatively – and perhaps a better approach since we do not possess starting and completions times of activities, we could also check for the absolute number of times someone activated the "Wait - User" state, instead of inspecting the duration. This provides the following list: Siebel, Pawel, Muthu, Brecht, Marcin, Fredrik, Andreas, Katia, Krzysztof, Emil. Just as for the previous questions, this count can also be aggregated at the case level. Follow-up questions by the process owner can thus quickly result in more fine-tuned analysis.

We close this question with Table 11, which provides an overview of the ten highest support teams, functional divisions, organization lines, countries and owners making most use of the "Wait - User" status, ordered by the number of times the associated actor executed a "Wait - User" status transition.

Table 10: Ten highest support team members spending most time in the "Accepted/Wait - User" state, ranked by median to account for exceptional outliers.

Owner First Name	Median Duration (Days)	Min Duration (Days)	Mean Duration (Days)	Max Duration (Days)	Total Duration (Days)
Gregor	29.38	28.87	29.38	29.89	58.76
Gabriel	27.52	0.01	27.52	55.03	55.05
Brice	21.10	2.98	21.10	39.23	42.21
Reinier	20.12	3.92	112.61	313.78	337.82
Susanne	18.07	0.00	29.20	78.02	175.20
Maurice	17.45	0.00	17.45	34.90	34.90
Earl	12.02	0.00	44.66	329.01	5761.26
Suliman	11.17	0.00	11.17	22.33	22.34
Bengt	9.57	0.00	9.57	19.14	19.14
Shery	8.46	0.03	8.46	16.89	16.92

Table 11: Ten highest support teams, functional divisions, organization lines, countries and owners making most use of the "Wait - User" status.

Support Team (Nr. of Events)	Functional Division	Organization Line	Country	Owner First Name
G97 704	V3_2 1930	Org line C 2783	Sweden 1227	Siebel 173
G96 226	A2_1 803	Org line A2 847	Poland 1018	Pawel 123
D5 212	E_10 383	Org line B 355	India 787	Muthu 80
D8 194	(blank) 258	Other 138	Belgium 305	Brecht 75
G92 191	E_5 248	Org line V2 50	Brazil 225	Marcin 71
G230 2nd 190	A2_2 180	Org line G4 12	USA 225	Fredrik 69
S42 178	D_1 161	Org line G2 10	0 174	Andreas 63
S43 136	A2_4 94	Org line V5 7	China 87	Katia 62
S56 123	A2_3 54	Org line V1 4	France 59	Krysztof 56
D7 97	E_4 31	Org line V11 3	Japan 43	Emil 55

4.4 Question 4: Process Conformity per Organization

This question pertains to "Org line A2" and "Org line C": "In general the Volvo IT organization is spread in two organizations: Org line A2 and Org line C. It would be interesting to see how conform or how much in line every organization is with the incident and problem management processes."

This is an open-ended question for which more domain knowledge must be gathered before it can be answered in a satisfactory way. The main question is what is meant with "in line". One particular perspective was already focused upon when dealing with the first question: which of these two organization lines perform most case escalations to higher level support teams?

When being "in line" is related to case duration, we can derive the mean case durations for "Org line A2" and "Org line C" respectively: 21 days versus 10 days (for the "incidents" log). When SLA's are set on total running time of case, the outcome might, however, be different. Consider for example an SLA determining that incident handling cases may take at most 7 days (one week). Now "Org line A2" violates this 1494 while "Org line C" performs worse: 3529.

Finally, it is also possible to view "in line conformance" in terms of a prescriptive process already in place. Ticketing based systems such as this one, however,

are not often subjected to such designed processes, as is also evidenced by the unstructured nature of the full process map with no filtering applied. As a proof of concept, however, let us assume a designed process as shown in Fig. 16. We will only make use of "Status" transitions to allow for enough freedom for the sake of this example. We assume that cases must start with an "Accepted" or "Queued" status. Both these statuses can repeat, and it is possible to go from one state to another as well. Finally, cases must be ended with a "Completed" state following "Accepted".

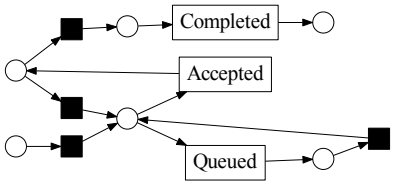


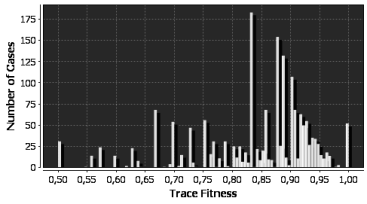
Fig. 16: An example designed prescriptive "to-be" process model.

We load this model into ProM in order to perform control-flow based conformance analysis. We use the alignment based conformance checking plugin by Adriansyah et al. [13–15]. Fig. 17 shows the results of the conformance analysis. Inspecting the trace fitness distributions (a measure of being in line with the prescribed process), we can derive that "Org line A2" is less conform (relatively) than "Org line C". Keep in mind, of course, that we have used an example model. By comparing the "as-is" situation with the "to-be" model, it becomes possible to implement changes in order to try to fix problem areas.

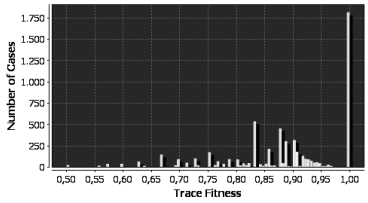
5 Conclusions

This report presented the results we uncovered related to the analysis of the Volvo IT data set for the Third International Business Process Intelligence Challenge (BPIC'13). An open-minded exploratory surveyance of the data set was performed, followed with a deeper analysis towards answering the specific questions posed by the process owner.

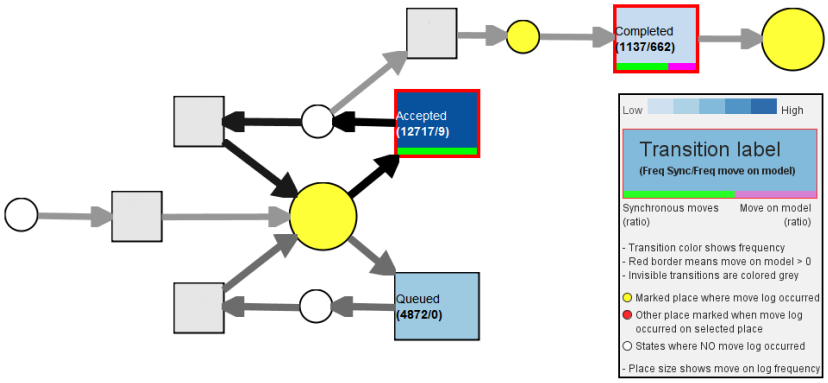
To do so, we have applied both traditional data analysis tools and process-oriented techniques. As evidenced by this report, there exists a gap between these two categories of tools, causing difficulties when filtering, querying, exploring and analyzing event based data sets, as is also put forward as one of the challenges listed in the process mining manifesto [2]. Therefore, we have utilized a hybrid approach, combining and switching back-and-forth between (mainly) R and Disco.



(a) Trace fitness distribution: "Org line A2".



(b) Trace fitness distribution: "Org line C".



(c) Aggregated overview highlighting conformance issues.

Fig. 17: Conformance analysis results for "Org line A2" on example process model.

Acknowledgment

We would like to thank the KU Leuven research council for financial support under grand OT/10/010 and the Flemish Research Council for financial support under Odysseus grant B.0915.09. We thank Véronique Van Vlasselaer for her insights regarding R.

References

[1] van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: Prom: The process mining toolkit. In de Medeiros, A.K.A., Weber, B., eds.: BPM (Demos). Volume 489 of CEUR Workshop Proceedings., CEUR-WS.org (2009)

[2] van der Aalst, W.M.P.: Process mining manifesto. In Daniel, F., Barkaoui, K., Dustdar, S., eds.: Business Process Management Workshops (1). Volume 99 of Lecture Notes in Business Information Processing., Springer (2011) 169–194 (Authors ommited.).

[3] Gabadinho, A., Müller, N.S., Ritschard, G., Studer, M.: Traminer: a toolbox for exploring sequence data
[4] Wang, H., Zaniolo, C.: Atlas: A native extension of sql for data mining and stream computations (2003)
[5] Law, Y.N., Wang, H., Zaniolo, C.: Query languages and data models for database sequences and data streams. In: Proceedings of the Thirtieth international conference on Very large data bases - Volume 30. VLDB '04, VLDB Endowment (2004) 492–503
[6] Law, Y.N., Wang, H., Zaniolo, C.: Relational languages and data models for continuous queries on sequences and data streams. ACM Trans. Database Syst. **36**(2) (June 2011) 8:1–8:32
[7] Sadri, M.R.: Optimization of Sequence Queries in Database Systems. PhD thesis, University of California, Los Angeles (2001)
[8] Sadri, R., Zaniolo, C., Zarkesh, A., Adibi, J.: Optimization of sequence queries in database systems. In: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. PODS '01, New York, NY, USA, ACM (2001) 71–81
[9] Sadri, R., Zaniolo, C., Zarkesh, A.M., Adibi, J.: A sequential pattern query language for supporting instant data mining for e-services. In: Proceedings of the 27th International Conference on Very Large Data Bases. VLDB '01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 653–656
[10] Kleene, S.C.: Representation of events in nerve nets and finite automata. In Shannon, C.E., McCarthy, J., eds.: Automata Studies. Princeton University Press (1956) 3–40
[11] van der Aalst, W.M.P., de Beer, H.T., van Dongen, B.F.: Process mining and verification of properties: An approach based on temporal logic. In: OTM Conferences (1). (2005) 130–147
[12] Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible heuristics miner (fhm). In: CIDM, IEEE (2011) 310–317
[13] Adriansyah, A., Sidorova, N., van Dongen, B.F.: Cost-based fitness in conformance checking. In Caillaud, B., Carmona, J., Hiraishi, K., eds.: ACSD, IEEE (2011) 57–66
[14] Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.: Alignment based precision checking. In Rosa, M.L., Soffer, P., eds.: Business Process Management Workshops. Volume 132 of Lecture Notes in Business Information Processing., Springer (2012) 137–149
[15] van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery **2**(2) (2012) 182–192