

BPIC 2014: Insights from the Analysis of Rabobank Service Desk Processes

Bruna Christina P. Brandão, Guilherme Neves Lopes, Pedro Henrique P. Richetti

Department of Applied Informatics - Federal University of the State of Rio de Janeiro,
Rio de Janeiro, Brazil

{bruna.christina, guilherme.lopes, pedro.richetti}
@uniriotec.br

Abstract. Process mining techniques allow knowledge extraction from events stored by information systems. For this challenge, four datasets related to Service Desk business processes were analyzed in order to discover and present some useful information about their behavior. The work has been done by looking to the process in different perspectives such as: conformance, social networking and a declarative way. The main objective is to deliver some interesting insights that may help process managers to improve their business processes.

Keywords: business process intelligence · service desk · process mining

1 Introduction

Process mining is a new discipline that links the disciplines of data mining and process intelligence. Process mining is the discovery of processes, their analysis and the verification if the process is efficient and how it can be changed to improve the business. The [IEEE Task Force, 2012] wrote a manifesto presenting the main concepts of process mining.

The process mining is executed through the analysis of event logs recorded on data in which the information has been made during the process execution. Therefore, the discovery from a process by mining event logs is called as-is process, as the process is really executed, being different from how they were modeled (to-be). But the process mining techniques not only presents discovery processes, also presents techniques for analyzing process performance and compliance techniques, to verify if the execution of the process is consistent with constraints and business rules [Dumas et al., 2013].

One approach which uses process mining is the analysis of process performance. The analysis is based on four dimensions: Time, Cost, Quality and Flexibility. The time dimension verifies the lifecycle of activities and the waiting time to be executed. The cost dimension is analyzed from the breakdown of prices present in logs. The quality is analyzed by the perception of the same activities performed more than once in a row in the same instance of a case, concluding that there is a need for a re-work

on an activity perhaps indicating a change of activity. And flexibility is the degree of variation the process allows, i.e. the process is checked to see if it is more or less flexible than desirable [Dumas et al., 2013].

Another approach is the verification of conformity, in which there is exclusivity constraint checks (activities that exclude other - e.g. activity request accepted and rejected request), the order of activities (e.g. rent equipment before checking availability) and of obligation (mandatory activities for any purpose of the organization - e.g. rent review application with the aim of controlling the cost). Also check business rules, whether the terms of the processes are fulfilled, e.g. if there is a purchase at a value higher than an established price, the acquisition must have confirmation from the manager [Dumas et al., 2013]. All the analysis from each of the dimensions assists in the decision-making of the process.

The rest of this work is structured as follows. Section 2 presents the dataset characteristics. Section 3 shows analysis of interactions that causes incidents. Section 4 presents a bottleneck analysis. Section 5 describes some insights from the variants of the Incident Treatment process. Section 6 shows a social view based on team roles. Section 7 demonstrates how a declarative approach may deliver some useful information about processes behavior and Section 8 concludes the report.

2 Dataset Characteristics

It was available four dataset presenting the process from Rabobank Group ICT. The dataset revolve around the service desk processes with Interaction Management, Incident Management and Change Management. The fourth dataset is a specification of the activities done in the incidents process. To understand how the dataset relate to each other, it was made a map of the relationship between them and Figure 1 represents it.

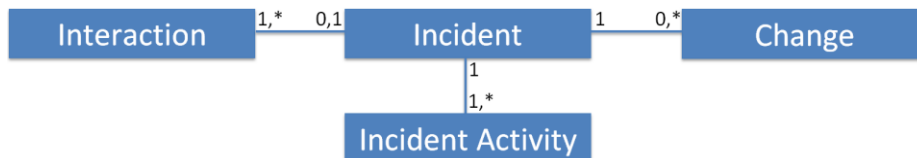


Fig. 1. Relationship between datasets.

Table 1 shows the attributes provided in the dataset and shows some rough information about the quantity of data to be analyzed. From this table is possible to take some conclusions, like the number of different incident id from the incident activity dataset is bigger than the number of different incident id from incident management dataset. This observation shows that there are incidents records in the incident activity which it weren't record in the incident database or that it were record with the wrong id. Another observation is about the handle time between interaction and incident. It can be notice that an incident takes a lot of more time to be handle than an interaction (average of 444,6673 seconds to 92491,01 seconds).

Table 1. Information about all dataset attributes.

Field	Interaction Management	Incident Management	Change Management	Incident Activity
CI Name (aff)	4153	3019	10193	X
CI Type (aff)	14	13	13	X
CI Subtype (aff)	67	65	74	X
Service Comp WBS (aff)	289	274	286	X
Interaction ID	147004	43060	X	46444
Incident ID	41835	46606	X	46616
Status	2	2	X	X
Impact	5	5	X	X
Urgency	6	6	X	X
Priority	5	5	X	X
Category	6	4	X	X
KM number	2360	1825	X	1825
Open Time (first)	09/09/2011 09:23	05/02/2012 13:32	X	X
Close Time (last)	31/03/2014 22:47	31/03/2014 22:47	X	X
Closure Code	24	14	X	X
First Call Resolution	2	X	X	X
Handle Time (secs)	444,6673	92491,01	X	X
Reopen Time (first)	X	10/04/2013 09:15	X	X
Resolved Time (last)	X	31/03/2014 22:47	X	X
Alert Status	X	1 (closed)	X	X
# Reassignments	X	46	X	X
#Related Interactions	X	370	6	X
# Related Incidents	X	63	279	X
# Related Changes	X	9	X	X
CI Name (CBy)	X	3652	X	X
CI Type (CBy)	X	14	X	X
CI Subtype (CBy)	X	63	X	X
ServiceComp WBS (CBy)	X	275	X	X
Change ID	X	232	18000	X
Change Type	X	X	240	X
Risk Assessment	X	X	3	X
Emergency Change	X	X	2	X
CAB-approval needed	X	X	2	X
Planned Start (first)	X	X	01/06/2011 07:00	X
Planned End	X	X	20/02/2021 17:30	X
Scheduled Downtime Start (first)	X	X	21/12/2012 14:29	X
Scheduled Downtime End (last)	X	X	21/03/2015 01:00	X
Actual Start (first)	X	X	16/10/2012 13:09	X
Actual End (last)	X	X	21/03/2021 00:01	X
Requested End Date (last)	X	X	20/02/2028 17:00	X
Change record Open Time (first)	X	X	01/09/2011 09:13	X
Change record Close Time (last)	X	X	31/03/2014 23:53	X
Originated from	X	X	3	X
DateStamp (first - last)	X	X	X	07/01/2013 08:17 - 02/04/2014 20:08
IncidentActivity_Number	X	X	X	466737
IncidentActivity_Type	X	X	X	39
Assignment Group	X	X	X	242

2.1 Detail Incident Activity

It was identified that the accuracy of the date stamp field is in seconds, and that certain activities in sequence end up being recorded in the same time. The problem found by this limitation of precision is that the sequence of activities "Closed" and "Caused

By CI" when occurring in the same time, the mining programs go on to create variations of the process with cases that the activity "Caused By CI" occurs after activity "Closed". In order to properly treat these variants, the log was modified to introduce a delay of 0.1 second in the activity "Closed" so that it always occurs after the activity "Caused By CI" as the specification of the fields: "When an operator resolves an Incident from an Incident, caused by the CI must be registered before the closing Incident-record". Without this treatment, the incident activity log contains 22632 variants. After adding 0.1s in the date stamp from "closed" activity, the number of variants was reduced to 21761.

2.2 Change Management

The fields Actual Start and End fields are empty in 3258 rows (11%) and they are equal in 8192 (27%) rows, from a total of 30.275 lines. These are the most accurate fields to consider the beginning and end of a change.

The fields Scheduled Downtime Start and Scheduled Downtime End are null in 29520 (97%) and 29531 (97%) rows, respectively. With these values being so significant, we can conclude that these fields are not normally used by the organization.

The Requested End Date field is desired by the applicant to implement the change. This date is regarded by Change Management but other factors such as availability of resources and technical constraints impact on the actual date of implementation, so this field cannot accurately indicate the end of the implementation of the change.

The Change record Open Time and Change record Close Time fields deal with the record of the change in HP Service Manager 9 tool. Not necessarily, the date of registration has connection with the date of the implementation. This is possible because the record may have been opened ahead and the change planned for a later date. The same thought is valid for closing the record, which can occur after implementing the change at any time.

The Planned Start field does not contain null values and Planned End field has 43 (0%) lines with null value. These fields have the same value in 2177 lines (7%), in despite of being complete in the log, these fields deal with the planning and may not accurately reflect the actual period of the implementation of the change. Considering only the rows where the values are not null, the differences between the planned date and the actual dates (current) are (values in days) found in the Table 2. Although the average is less than two days, the standard deviation is high.

Table 2. Difference between Planned and Actual date.

	Planned Start – Actual Start	Planned End – Actual End
Average	-2,04	0,85
Standard Deviation	36,86	32,43

It will be consider the end date that the values are not null of "Actual End" to the evaluation of the increased volume in the service desk after the implementation of a change. This results in a total of 27014 records (89%). The field "Actual End" has

twelve records with future date, after May of 2014. These cases will be disregarded from the analysis and can be seen in Table 3.

Table 3. Future date in Actual End.

CI Name (aff)	CI Type (aff)	CI Subtype (aff)	Service Component WBS (aff)	Change ID	Planned Start	Planned End	Actual Start	Actual End
HMD000058	hardware	MigratieDummy	WBS000258	C00007614	04-dez-13	04-dez-13	04-dez-14	04-dez-14
HMD000058	hardware	MigratieDummy	WBS000258	C00007614	04-dez-13	04-dez-13	04-dez-14	04-dez-14
SBA000308	application	Server Based Application	WBS000332	C00011299	17-jan-15	17-jan-15	17-jan-15	17-jan-15
DBI000138	database	Instance	WBS000253	C00013238	31-jan-14	04-fev-15	31-jan-14	04-fev-15
DBI000140	database	Instance	WBS000253	C00013238	31-jan-14	04-fev-15	31-jan-14	04-fev-15
DBI000162	database	Instance	WBS000253	C00013238	31-jan-14	04-fev-15	31-jan-14	04-fev-15
DBI000164	database	Instance	WBS000253	C00013238	31-jan-14	04-fev-15	31-jan-14	04-fev-15

2.3 Interaction

Figure 2 shows that the most interactions were ended with the code “other”.

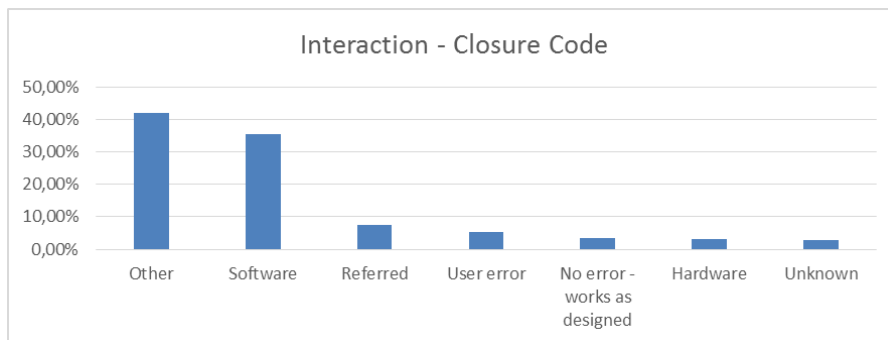


Fig. 2. Distribution of interactions by closure code.

Figure 2 also shows the distribution of the main codes for closure in the category. Figure 3 presents only categories with more than 2% of records of the interactions. The majority of the interactions it's in the "other" and "software" category.

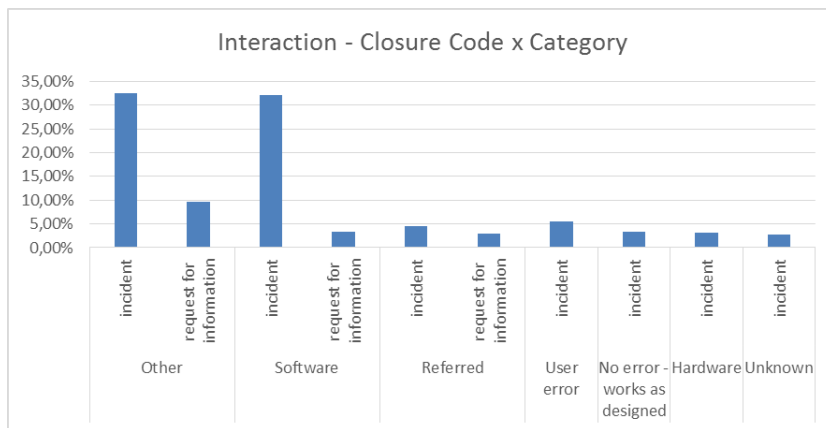


Fig. 3. Distribution of interaction by closure code and category.

2.4 Incident

Figure 4 shows that the most incidents were ended with the code "other". It also shows the distribution of the main codes for closure in the category from incident dataset. Figure 5 presents 96% of the incidents that are focus in 12 subtypes. The majority of the incidents area in the "other" and "software" category.

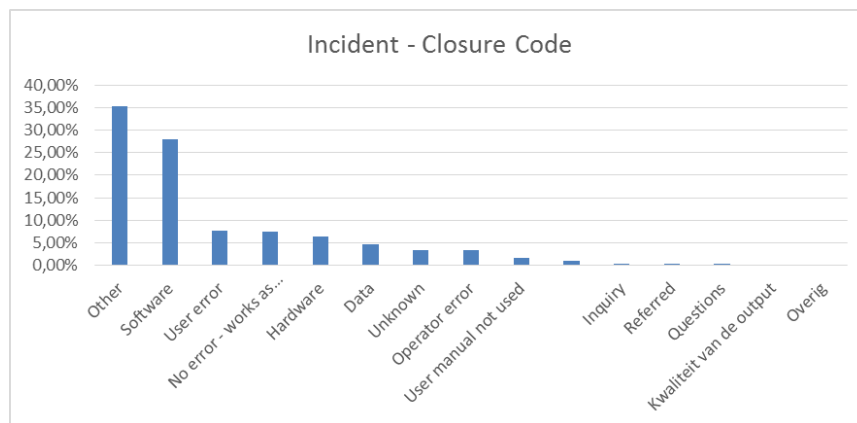


Fig. 4. Distribution of incidents by closure code.

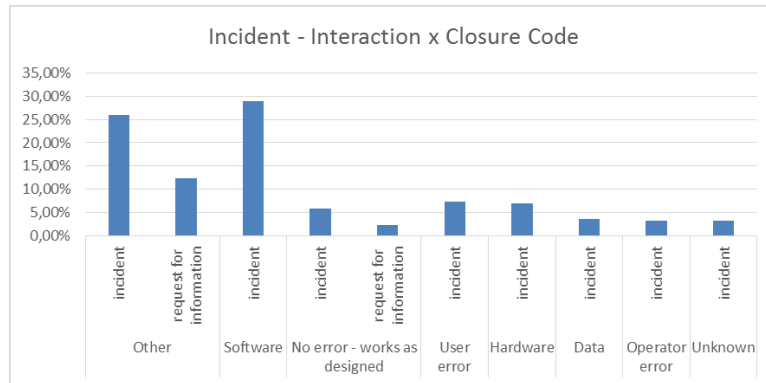


Fig. 5. Distribution of incidents by closure code.

2.5 Time Distribution

It was made an analysis of the time distribution and it was observed that the average behavior from the interactions is similar to the incidents throughout the period of a day (07h to 18h). Table 4 shows the correlation between interaction, incident and change. Figure 6 and Figure 7 show the hour distribution by day of interaction, incident and change data. Figure 6 shows an absolute distribution and Figure 7 shows a relative distribution.

Table 4. Time distribution Incident vs interaction.

		Correlation
Interaction	Incident	0,9944
Incident	Change	0,9346
Change	Interaction	0,9139

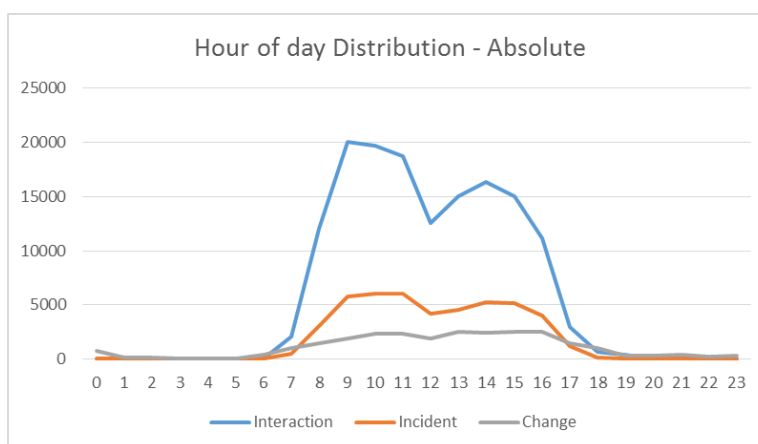


Fig. 6. Distribution per hour - absolute.

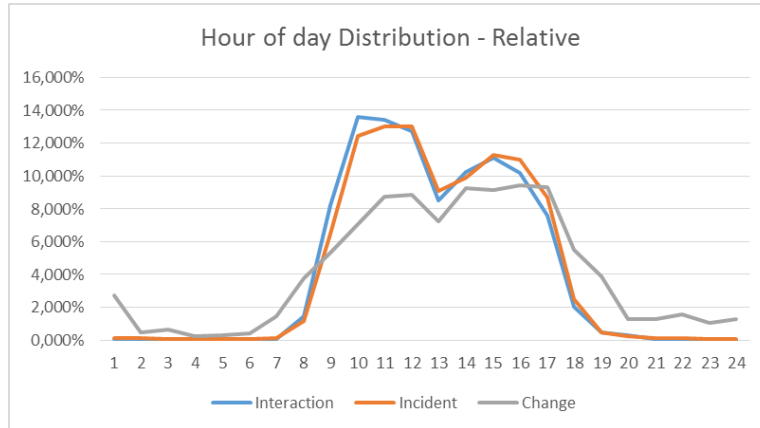


Fig. 7. Distribution per hour - relative.

3 Interactions that causes incidents

It was identified the service components that had the most interactions becoming incidents for the IT Operation. It was selected the service components that correspond to 80% from the incidents opened by interaction, with a total of 32 service components. Through the selected services components, it was identified the respectively closure code of the incidents that don't have related changes. Table 5 shows an example with the service component WBS000073 selected.

Table 5. Service Component with most interactions with incidents.

Service Component WBS (aff)	Closure Code	ContarDeIncident ID
WBS000073	Data	802
WBS000073	Hardware	4
WBS000073	No error - works as designed	830
WBS000073	Operator error	812
WBS000073	Other	3716
WBS000073	Referred	17
WBS000073	Software	4697
WBS000073	Unknown	470
WBS000073	User error	1150
WBS000073	User manual not used	383

This analysis results in 4697 incidents with the closure code "Software" that it didn't have an associate change. This could indicate that the problem is known and that it's not sent to treatment, which can generate extra work for the second level teams.

It's also not desirable that the incidents treat problems cause by user. For example, the presence of incidents with the closure code being "User error" or "User manual not used" should be avoided. And it can be resolve with user training and this avoids that the operation team needs to investigate situations which are not configurable as real incidents.

4 Bottleneck analysis

To study bottleneck it was chosen two datasets, incidents and change. There is a relation between this dataset using the attribute change id. It was excluded results that the change id is null. The incidents dataset has 46809 records but only 868 generate changes. Using the attribute change id as a case id, it results on a process view that shows how changes affect incidents.

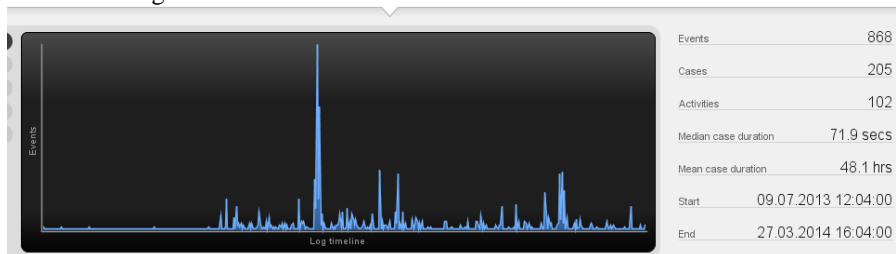


Fig. 8. Disco statistics.

Figure 8 shows statistics obtained using disco to mining process. There are 205 cases which mean that are 205 unique change identifications. 102 activities that was a join from the item configuration id plus item configuration subtype and plus the change type. This concatenation makes a unique id with the capability to represent the type of change on specific subtype and type of configuration item. And 868 events which are all the activities executed in this cases.

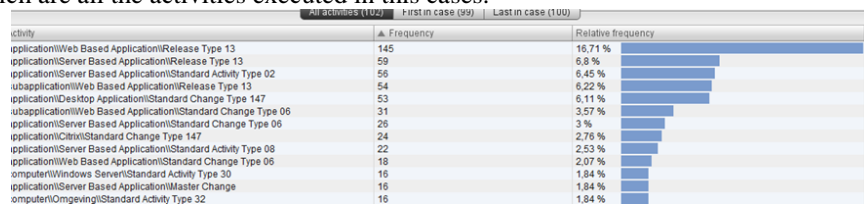


Fig. 9. Most frequent activities.

Analyzing the most frequent activity that it occurs 145 times, one of strategies to follow is to simplify the model with a filter of the most frequent activity. The process resulting after apply the filter is showing in Figure 10.

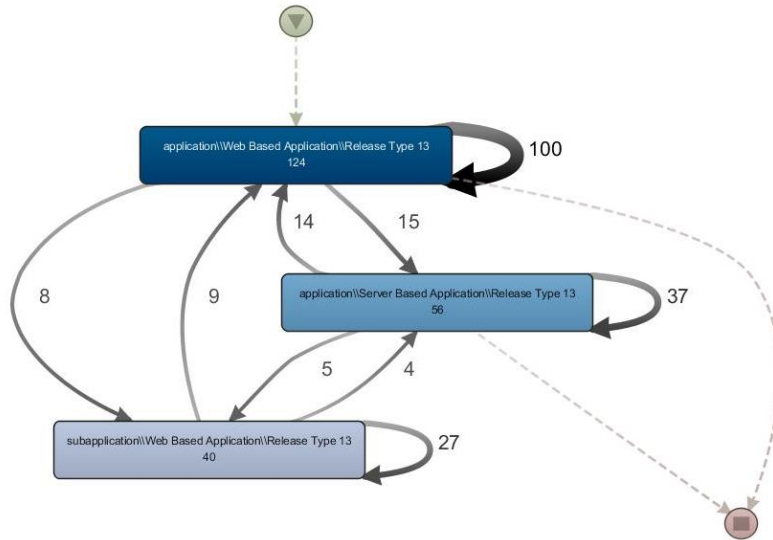


Fig. 10. Process filter by activity.

The process in Figure 10 is a bottleneck Picture, a sequence of looping. Solving this looping is a good strategy to minimizing time and money. Rework is not good because it adds to the workload (and costs) of the company, and it delays the process completion time for the customer, and because—due to the extra effort—it often impacts the completion times for successive cases as well [Fluxicon, 2014].

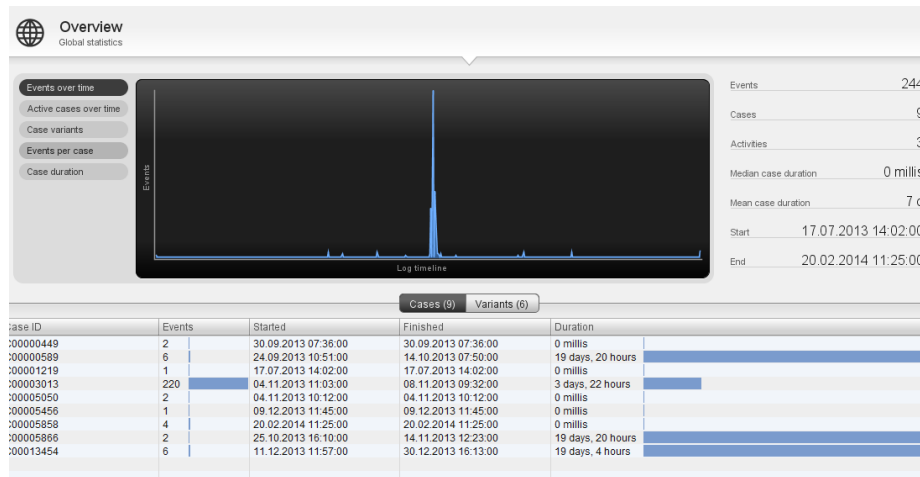


Fig. 11. Statistics of filtered events.

This 3 activities affects only 9 cases, in this cases only 4 are abnormal. The suggestion is verify these 4 cases overtime and redesign the process. The change id

“C00003013” has 220 events, this number of events is greater than other cases, on changes C0000589, C00005866 e C00013454 the activity duration is greater than other cases.

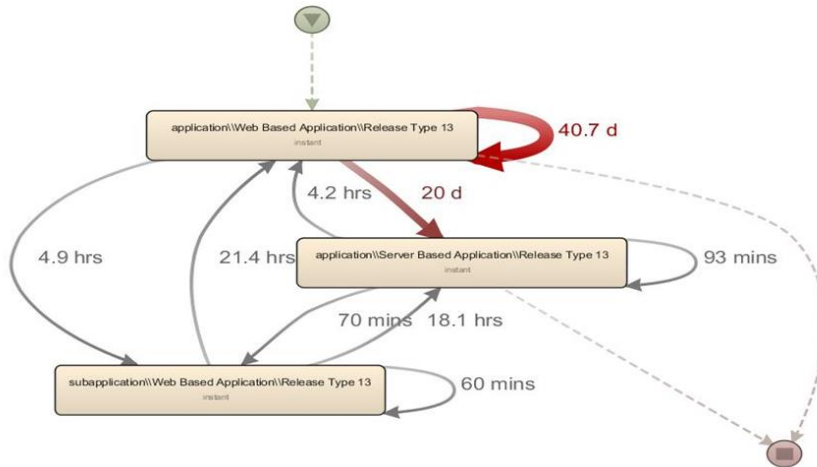


Fig. 12. Process map for rework total time for Release Type 13.

The time spent on rework is very important, the special attention on flow that total time is 40.7 days and the transition with 20 days. These flows must be audited.

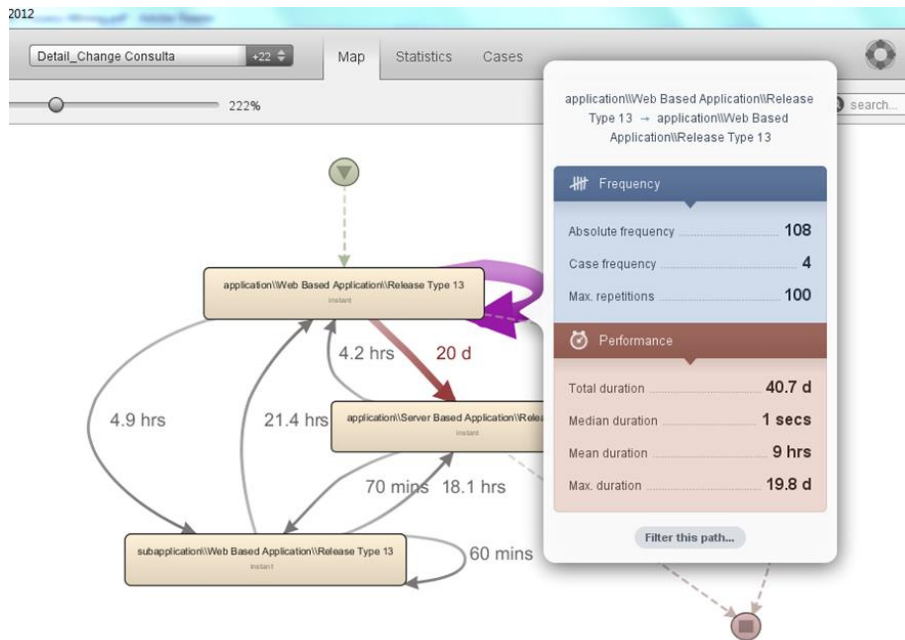


Fig. 13. Detail on rework total time for Release Type 13.

Another relevant activity is type 147 the Picture represents a bottleneck, affect 3 types of type and subtype. The number off loops suggest rework.

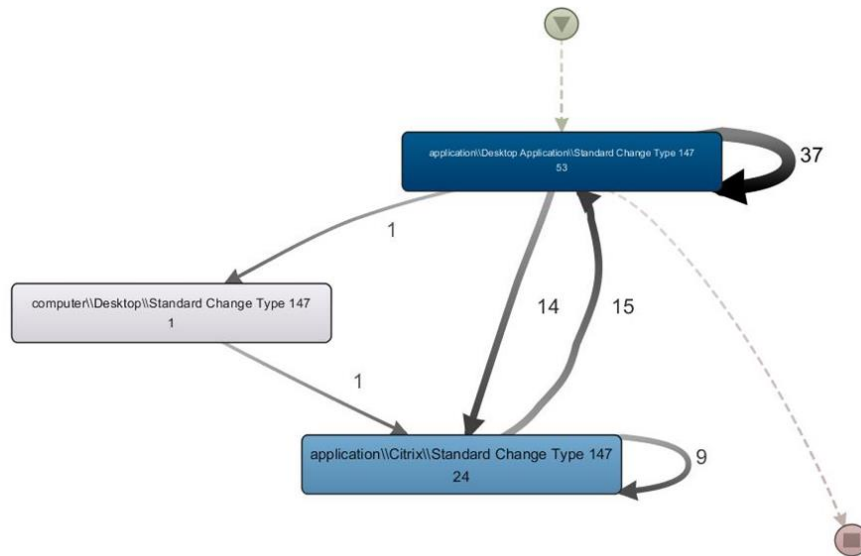


Fig. 14. Rework on change type 147.

The change was generated 78 incidents with only 3 differences Configuration Items with a duration of 10 days, shows that the problem has reached a large number of users, with a big impact. If we look more closely, we see the following paragraphs.

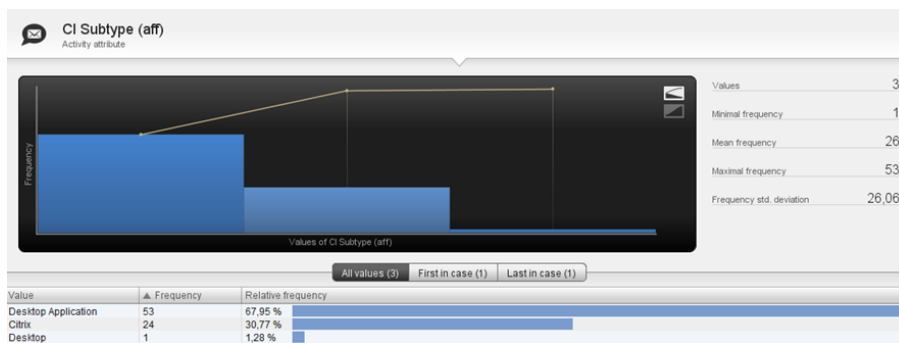


Fig. 15. Configuration Item Statistics.

The Change Type 6 is an important change, analyzing using declarative methods, we find an enhanced of incidents when type 6 was implemented. Using Fuzzy method, the Figure 16 shows a bottleneck.

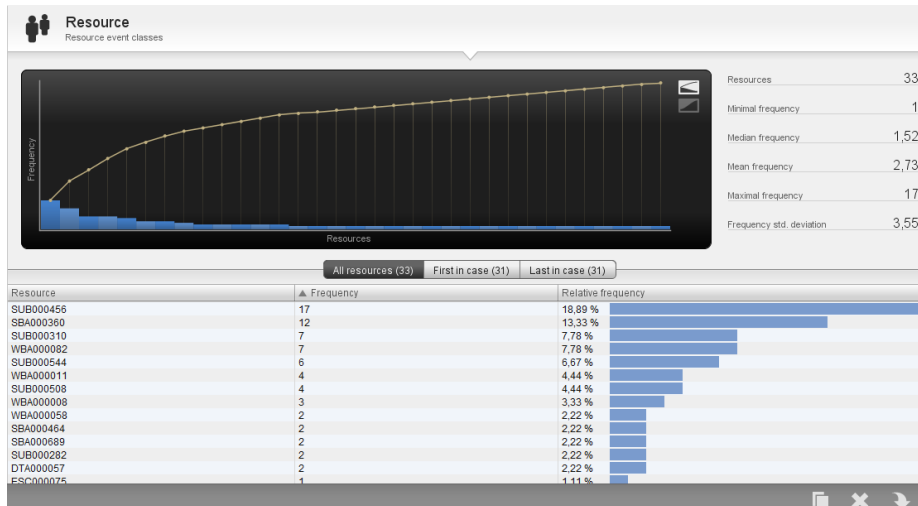


Fig. 19. Most affected configuration items.

5 Variants of the Process From the Incident Treatment

To identify the main process from the incident treatment, it was made a ranking from the Detail Incident dataset with the field category, configuration item type, and configuration item subtype and service component. The Figure 20 shows the rank from the category field, which the category of “incident” has more than 80% of frequency.

Value	Frequency	Relative frequency
incident	37748	80,99 %
request for informat...	8846	18,98 %
complaint	11	0,02 %
request for change	1	0 %

Fig. 20. Rank of type of category from Incidents.

After choosing to look only to the most category (“incident”) present in the log since the second category (“request for information”) more frequent is not significant to the purpose of finding out the main process more frequent in the incidents. It was chosen to do a rank of CI Type and CI Subtype within the incidents that have the category “incident”. The same was made to the service component. The Figure 21, Figure 22 and Figure 23 show the first three most frequent type, subtype of CI and service component respectively.

Value	Frequency	Relative frequency
application	25370	67,21 %
subapplication	6788	17,98 %
computer	3544	9,39 %

Fig. 21. Rank of CI type.

Value	▲ Frequency	Relative frequency
Web Based Application	14650	38,81 %
Server Based Application	12574	33,31 %
Desktop Application	3499	9,27 %

Fig. 22. Rank of CI Subtype.

Activity	▲ Frequency	Relative frequency
WBS000073	13057	34,59 %
WBS000091	2431	6,44 %
WBS000263	1401	3,71 %

Fig. 23. Rank of Service Component.

After this decision to work with the most frequent category, CI type, CI subtype and service component. The dataset was include in the Oracle database and after the join from the dataset detail incident and incident activity to show only the data required, it was exported to a csv file to import in Disco to find out the variants of the process. The Figure 24 shows the process with less than 1% of variants and 32% of cases. The Figure 25 shows the same dataset with less than 11% of variants, but with 51% of the cases.

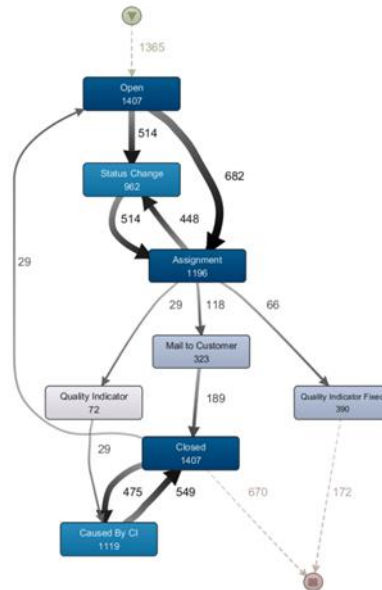


Fig. 24. Incident process 1% variants.

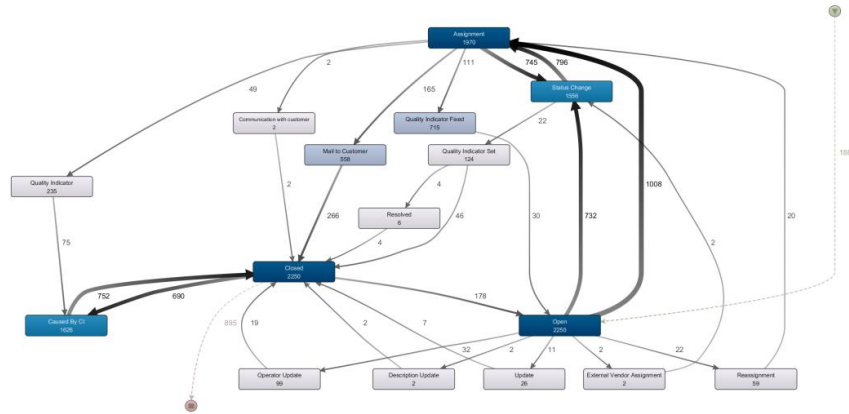


Fig. 25. Incident process 11% variants.

This data show that the most frequent incident process for the same category, CI type, CI Subtype and service component doesn't follow the same process. Only 32% of the cases follow the process in Figure 24 and to have half of the cases (51%), the process has 11% of variants like Figure 25, resulting in a process with a lot of paths. The incident process specialist should analyzed both process and discover if the process in Figure 24 is too simple to really solve the incident or if the teams doing this kind of incident don't know the simple path.

The same was made with the others service component ("WBS000263" and "WBS000091"). And the conclusion from this service component with the incident that has the category as incident, it is that the service component WBS000073 has the item configuration type with the same subtype. That is the type application has Web Based Application, Server Based Application, Client Based Application, Standard Application subtypes, and the type subapplication has two of the application subtype (Web Based Application, Server Based Application). This division didn't have to be made, it's unnecessary. With de service component WBS000263 it was discover that it only has one type and one subtype of configuration item that cause incidents (type application and subtype Server Based Application). This suggests having a better training and testing this service component in the configuration item that has its type as application. The same can be concluded from service component WBS000091, the subtype from configuration item that has its type as application is Desktop Application. This also can suggests having a better training and testing in the configuration item that has the subtype as Desktop Application.

6 Social Mining

The Incident Activity dataset presents information about the teams assigned to execute IT operations activities over the incidents. This log contains 39 different activities realized by a total of 242 distinct teams. The question to be answered is if this

high number of teams can be reduced. Our approach aims to search for teams having the same behavior over some dimension of the log, e.g. activity types or service components. It was defined a similarity measure between pair of teams that permit building a graph view of teams relatedness. For this experiment the following tools were used: MS Excel electronic datasheets, Gephi graph editor and Java programming language.

It is known that is possible to discover social networks that show the handover between the resources in a business process. ProM has a social Miner plugin able to do this work. This view allows the analysis of the interactions among the teams. We propose a complimentary perspective that looks for the similarity between the teams that can help managers to take decisions about aggregating teams with similar behavior. The idea is to create a matrix which lines contain the teams and the each column is a value from a selected dimension, e.g. for the activity dimension each column represent an activity type. For each team, if that team executed an activity it is marked with a "1", else it is marked as "0". This way it is possible to create a binary sequence where each bit position represents an activity. Figure 26 shows an example of this matrix. Inspired by the sequence alignment concept [Kleinberg, 2005] it is possible to determine similarity between a pair of teams by counting the mismatches among their binary sequences.

	Affected CI Change	Alert stage 1	Analysis/Research	Assignment	Callback Request	Caused By CI	Closed	Communication with customer	Communication with vendor	Contact Change	Description Update	Dial-in	External Update	External Vendor Assignment	Impact Char	Incid
TEAM9999	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
TEAM0008	0	0	0	1	1	1	1	1	0	1	0	0	1	1	1	1
TEAM0092	0	0	1	1	0	1	1	1	0	1	0	0	1	1	1	0
TEAM0033	0	0	0	1	0	1	1	1	0	1	0	0	1	1	1	1
TEAM0003	0	0	1	1	0	1	1	1	0	1	0	0	1	0	1	0
TEAM0010	0	0	0	1	0	1	1	1	0	1	0	0	1	1	1	0

Fig. 26. Example of binary sequences related to activities executed by each team.

The number of mismatches is calculated applying a XOR operation between two binary sequences and after counting the number of resulting "1" bits. As it is necessary generate all team pair combinations, a Java program was written to support this calculus. In order to normalize the results, we defined the similarity grade by equation below:

$$similarity = \frac{(Number\ of\ Activities) - (Number\ of\ Mismatches)}{(Number\ of\ Activities)}$$

The similarity grade values will range from 0 to 1 and the first value means a pair doesn't have common executed or non executed activities. The second value means a pair perform or does not perform the same activities. As an output from the Java pro-

gram, a set of pairs with their respective similarity value is generate, as seen in Figure 27. This information allows building an undirected weighted graph where each node relates to a team and the edges are weighted with the similarity for each pair.

```
TEAM9999; TEAM0008; 0. 71794873;  
TEAM9999; TEAM0092; 0. 7692308;  
TEAM9999; TEAM0033; 0. 6923077;  
TEAM9999; TEAM0003; 0. 64102566;  
TEAM9999; TEAM0010; 0. 6923077;|
```

Fig. 27. Output from similarity calculus between IT operation teams w.r.t. their performed activities.

Using Gephi a graph visualization was generated. By filtering the edge weight, a manager may define its range to view only the most interesting relations. As an example, Figure 28 shows the resulting graph from filtering only the maximal similarity value relatedness, i.e., the teams that performed or did not perform the same activities.

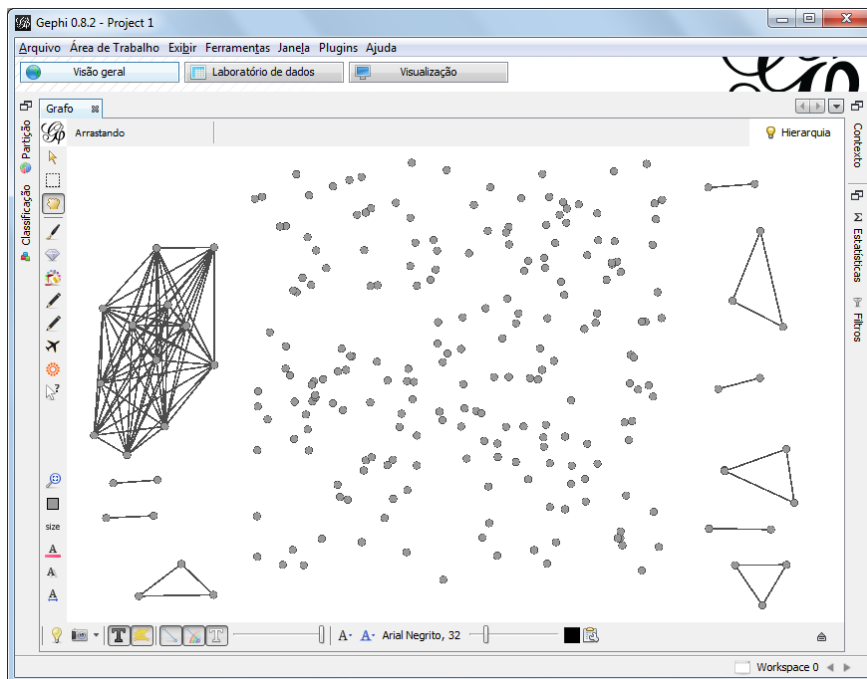


Fig. 28. General view showing all nodes and only the edges with similarity equal to one.

Using the filter set above, the result is a graph with ten groups. To improve the analysis all nodes that have no connections were excluded from the view. The resulting graph is shown in Figure 29.

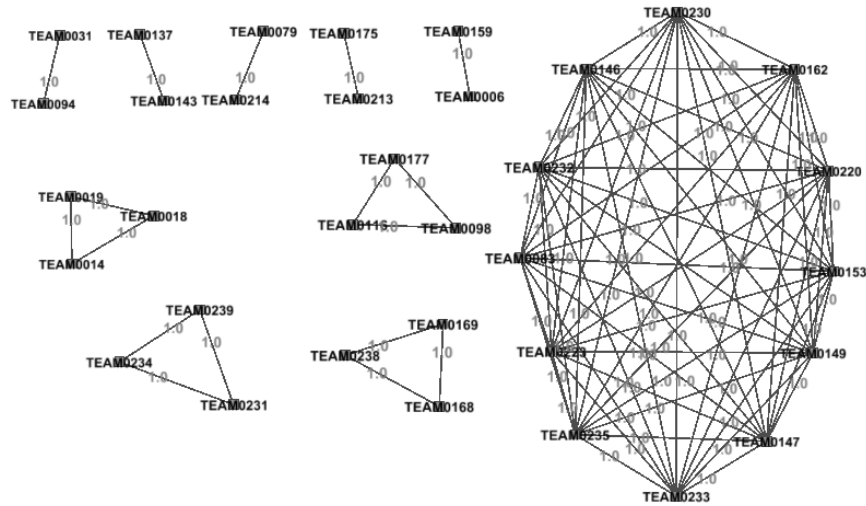


Fig. 29. Groups of teams with similarity equal to one.

Knowing activities nature and team roles, a manager may check if there are deviations or trends on teams operations. It may also points opportunities for team aggregation. Detailing the graph from Figure 29, there is a group with twelve activities. Looking for these teams in the matrix it is possible to identify all three activities performed by this group, they are: “Assignment”, “Reassignment” and “Operator Update”. This may denote these teams are responsible to designate the incidents or they are only repassing their received activities.

The activity similarity perspective has its relevancy, but there may exists situations where similar tasks are performed in different contexts, e.g. for different service components. So, it is also interesting to know the similarity between the service components handled by the operations teams. The same aforementioned analysis was performed for the service components.

Considering all the 275 service components present in the Incident Activity log, the team’s matrix was created and the binary sequences were generated. The similarity calculus was performed and the graph view was built. The first sight was the groups with maximum similarity, but due the high number of service components, only one group was formed, by TEAM0149 and TEAM0223. The manager, in a user-guided fashion, can reduce the similarity threshold. A 0.996 similarity threshold already allows building the graph view with some interesting relations. Figure 30 presents the resulting graph.

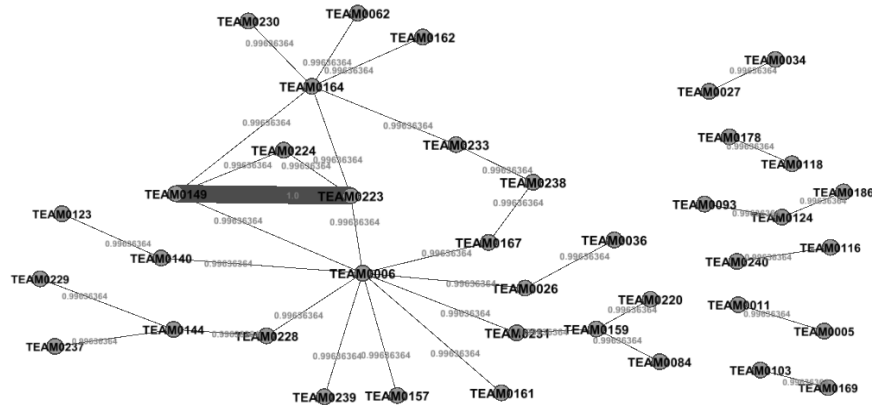


Fig. 30. Service components similarity for teams with a 0.996 threshold.

Again, the manager may verify the neighbors each teams has and check if it makes sense or if it looks like a deviation. The manager can also discover that some teams work with similar service components and then decide to group them.

7 Discovering Declarative Constraint to Identify Impact-Patterns

In order to identify patterns that correlates the implementation of changes and the workload in the Service Desk and IT Operations we propose the use of a declarative modelling language to identify temporal constraints that may reveal the aforementioned relation. A declarative modeling language respects an open world perspective, where everything that is not forbidden is possible. A declarative process model may represent the process behavior by a set of constraints that limits its possible executions in a circumstantial way.

There are some declarative process modeling languages available today, and due to the ease of use and tool support availability, we choose the Declare language [van Der Aalst, 2009], which is grounded on constraint templates modelled in linear temporal logic (LTL). A set of Declare constraints is presented in [Maggi, 2011]. The tool support is a great advantage of Declare, it has a modelling tool (CPN Tools) and a discovery and analysis environment by a set of plugins in the ProM framework. Although the ProM DeclareMiner exists, it is unfeasible for logs with a high number of traces because the exponential complexity of its discovery algorithm. As an alternative we used the UnconstrainedMiner [Westergaard, 2013] that uses a different approach for the discovery, by using regular expressions instead of LTL and some techniques to improve efficiency, such as: symmetry reduction, prefix sharing, super scalarity and parallelism. As a result, the UnconstrainedMiner output is a set of all possible constraints of the discovered declarative process model. To filter only the interesting behavior, we need to do some pre and post processing.

To run this experiment, the initial activity is the dataset preparation. For the first analysis we looked for the change implementation and the Service Desk workload, so we gathered information from de Change and the Interaction dataset. For the Interaction dataset we extracted the CI Name and counted the number of interactions open per day. For the Change dataset, we extracted every change and its respective Change Type and the CI Name. We wanted an event log with the CI Name as Case ID, the Change Types and Interaction Quantities as Activities, and the timestamp as itself. We noted that if we use each quantity as an unique activity, this number will be high. So we defined discrete intervals for activity number reduction that will also help focusing on the most interesting constraints. The intervals for interaction quantities are “<10”, “10-49”, “50-99” and “>100”. As we consider for interaction quantity a daily basis, we set the time information on the timestamp to the end of a day (23:59). Figure 31 shows the occurrence of daily interaction quantities on Interaction Dataset, there we can see the higher frequencies on lower quantities. This means that, in general, the standard daily workload for each CIs on Service Desk is low.

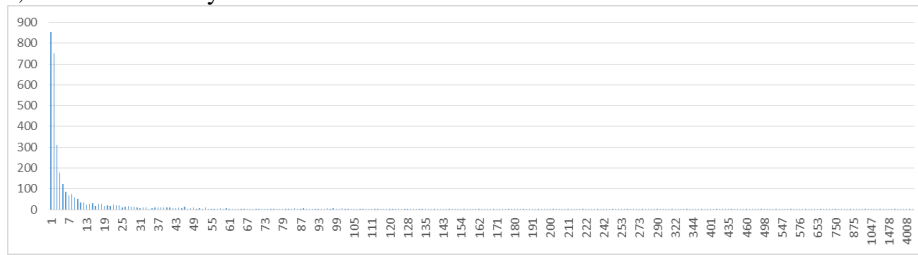


Fig. 31. Occurrence of daily interaction quantities on Interaction Dataset (y-axis denotes how many times a quantity occurred in the event log).

With the discrete value for quantities defined we built a new event log, merging changes and interactions, as seen in Table 6.

Table 6. Example of the generated event log.

CASE	TIMESTAMP	ACTIVITY
WSR000219	28/3/14 18:00	Standard Activity Type 54
WSR000219	28/3/14 23:59	<10
WSR000220	18/11/13 23:59	<10
WSR000220	28/3/14 18:00	Standard Activity Type 54
WSR000221	28/3/14 18:00	Standard Activity Type 54
WSR000221	31/3/14 23:59	<10
WSR000222	28/3/14 18:00	Standard Activity Type 54
WSR000223	28/3/14 18:00	Standard Activity Type 54

The produced log contained 33.879 events, 11.381 cases and 244 distinct activities. The next step is to mine declarative constraints using the UnconstrainedMiner. This mining step produced a huge output with thousands of declarative constraints, but we are looking for the most interesting constraints for this scenario: the chained ordered ones, because they will show when a change directly precedes or succeeds any quanti-

ty of interactions. So, we filtered only the chained ordered constraints, the constraints with no dependent support (meaning they weren't activated in the log) and the quantities "<10" and "10-49" were also removed from the results. We stay with 83 constraints that point to candidate Change Types that may have been impacted the workload on Service Desk. The top ten constraints are shown in Table 7. As we know the highest quantities are low on frequency, we did not expect they would have high support on the event log, but they still important because they cause the major impact on the Service Desk.

Table 7. Top ten constraints between Change Types and Interaction Quantities.

constraint	parameters	dependent
chain response	[[Standard Change Type 06], [>100]]	6
chain precedence	[[Standard Change Type 156], [50-99]]	5
chain precedence	[[Standard Change Type 06], [50-99]]	4
chain response	[[Standard Activity Type 03], [>100]]	3
chain precedence	[[Standard Activity Type 02], [50-99]]	3
chain precedence	[[Standard Change Type 156], [>100]]	3
chain precedence	[[Standard Change Type 06], [>100]]	3
chain response	[[Release Type 01], [>100]]	2
chain response	[[Release Type 01], [50-99]]	2
chain precedence	[[Standard Change Type 147], [50-99]]	2

We can think of declarative constraints as rules that shows the process behavior. As an example from Table 7, the first constraint says that every time an "Standard Change Type 06" occurred, directly after it more than 100 interactions were registered on the SD. With this rule in mind, we can go back to Disco software to detail which CIs presented this behavior. Using the follower filter, we can see 15 cases, which a "Standard Change Type 06" occurred, and immediately after it more than 100 interactions were opened on the SD. They are CBA000014, SBA000079, SBA000088, SBA000189, SBA000451, SBA000458, SBA000459, SBA000462, SBA000747, SBA000834, WBA000011, WBA000018, WBA000058, WBA000060 and WBA000133. Figure 32 shows part of a trace for WBA000133 confirming the previous identified behavior. The same detailed must be done for the other constraints in order to identify which change Types and CIs imply on the Service Desk workload.

Activity	Date	Time
Standard Change Type 06	26.01.2014	15:59:00
Standard Change Type 09	27.01.2014	11:43:00
50-99	27.01.2014	23:59:51
Standard Change Type 06	30.01.2014	14:45:00
Standard Change Type 06	30.01.2014	16:26:00
>100	30.01.2014	23:59:51
Standard Change Type 06	03.02.2014	08:45:00
Standard Change Type 06	03.02.2014	09:20:00
Standard Change Type 138	03.02.2014	11:50:00
>100	03.02.2014	23:59:51
Standard Change Type 06	10.02.2014	07:45:00
Standard Change Type 06	10.02.2014	07:51:00
Standard Change Type 06	10.02.2014	07:54:00
>100	10.02.2014	23:59:51

Fig. 32. Detail of the WBA000133 case that satisfies the chain_response(Standard Change Type 06, >100) constraint.

We did the same analysis for the correlation between changes and incidents. The only difference was the discretization levels: “<5”, “5-9”, “10-19” and “>20”. These levels were based on the occurrence of incidents day by day, as shown in Figure 33. The top ten constraints for more than 10 incidents per day are presented in Table 8.

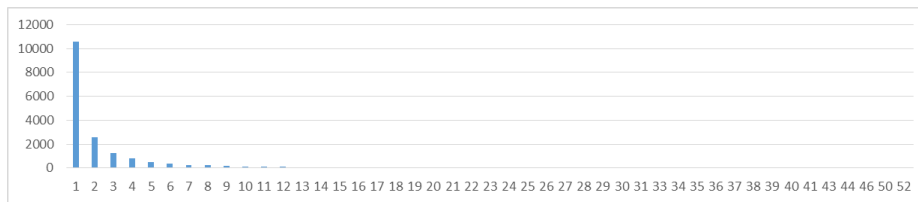


Fig. 33. Occurrence of daily incident quantities on Incident Dataset (y-axis denotes how many times a quantity occurred in the event log).

Table 8. Top ten constraints between Change Types and Incident Quantities with more than 10 incidents per day.

constraint	parameters	dependent
chain response	[[Standard Change Type 06], [>20]]	10
chain precedence	[[Standard Change Type 06], [10-19]]	7
chain precedence	[[Standard Change Type 156], [>20]]	7
chain precedence	[[Release Type 13], [10-19]]	6
chain precedence	[[Standard Activity Type 02], [10-19]]	5
chain precedence	[[Standard Change Type 06], [>20]]	5
chain precedence	[[Standard Change Type 156], [10-19]]	4
chain response	[[Standard Activity Type 02], [>20]]	4
chain response	[[Standard Activity Type 08], [10-19]]	3
chain precedence	[[Standard Change Type 42], [10-19]]	3

We have to consider a limitation of this approach: there may be some relations having a distant temporal relation, even though they have a direct following link between the activities, they may have no causal connection.

With this experiment, we aimed to show how declarative models could be used in practice, also providing useful insights for business process analysis. As the tool support for declarative process modelling is increasing, it is going to be easier to apply this perspective on real life problems.

8 Conclusions

This work has the purpose of helping process managers to have a clear view on their business process, showing them where some opportunities for improvement are. We focused on the search for impact-patterns among change implementations and the others service desk activities. Attention was also paid to the creativity challenge, where we presented a social network analysis and the use of declarative modelling to search for impact-patterns.

9 References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A. "Fundamentals of Business Process Management". Springer, 2013.
2. IEEE Task Force on Process Mining. "Process Mining Manifesto". Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011.
3. J. Kleinberg, E. Tardos. Algorithm Design. Addison Wesley, 2005.
4. Maggi, F. M., Mooij, A. J., van der Aalst, W. M. P.: User-Guided Discovery of Declarative Process Models. In: IEEE Symposium on Computational Intelligence and Data Mining, pp. 192-199. IEEE Computer Society (2011)

5. Van der Aalst, W. M. P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development* 23, 99-113 (2009)
6. Westergaard, M., Stahl, C., Reijers, H.A. UnconstrainedMiner: efficient discovery of generalized declarative process models. (External Report, BPM Center Report, No. BPM-13-28). BPMcenter.org, 28 pp. (2013)
7. Fluxicon. How to Identify Rework in Your Process, <http://fluxicon.com/download/Analyzing-Rework-With-Process-Mining.pdf> (2014)
8. IEEE Task Force on Process Mining: Process Mining Manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM Workshops, LNBIP*, vol. 99, pp. 169-194. Springer Berlin Heidelberg (2012)