Using Sequence Classification to Label Behavior from Sequential Event Logs

Scott Buffett, Bruno Emond, Cyril Goutte
National Research Council Canada

July 14, 2014

Abstract

This paper reports on findings resulting from our research conducted for the 2014 Business Process Intelligence (BPI) Challenge. The goal of this year's challenge was to identify predictive patterns in the data, provided by Rabobank's ICT operations unit, that could be used to determine whether service calls and incidents were likely to increase or decrease as a result of any change implemented in their systems. We took two approaches: a macro approach where conclusions were drawn via analysis of aggregated data from the entire set, and a micro approach where sequence classification was used to uncover predictive patterns in the data for individual configuration items. Each approach addresses the challenge questions from different levels, and offers insight on the findings from each point of view that can be applied to improve company business processes and decision making in the future.

1 Introduction

The Business Process Intelligence Challenge, held in conjunction with the Business Process Intelligence (BPI) Workshop, and the International Conference on Business Process Management (BPM), is an annual competition targeted toward business process mining researchers and practitioners. Each year, a real-life data set is anonymized and made available to the participants, as well as some general challenge questions or goals intended to motivate the analysis. This year, data sets were provided by the ICT operations unit for Rabobank, a banking and financial services institution headquartered in the Netherlands. The goal of this year's challenge was to identify predictive patterns in the data that could be used to determine whether service calls and incidents, initiated by users with technical difficulties, were likely to increase or decrease as a result of any change implemented in their systems.

We looked at the data from two different points of view. First, a high-level aggregated approach to the analysis was taken in order to discover an overall view of the business process and to identify any general trends in the data. Rabobank's challenge questions were addressed taking this overall view and conclusions that are widely generally applicable are drawn. Second, a more microscopic approach is taken where we analyze trends and patterns in events and activities relevant to each of the more heavily utilized configuration items. The methodology here employed techniques from *sequential pattern mining* and *sequence classification* to discover sequential patterns in the data that were shown to be predictive and thus useful in answering the challenge questions from a lower level.

The paper is organized as follows. After providing some more in-depth background on the challenge itself, including specifics on this year's data and challenge questions, we explore the challenge questions from the high-level, holistic point of view. This study is then followed by a lower-level analysis, where techniques from sequence mining are introduced and subsequently utilized to answer the challenge questions for specific configuration items. Finally, we conclude with some closing remarks.

2 Business Process Intelligence Challenge

2.1 Background

This paper is an entry into the fourth annual Business Process Intelligence (BPI) Challenge. The goal of the challenge is to promote the field of process mining and to "give both researchers and practitioners the opportunity to do process mining analyses on real-life data" [3]. Each year, participants are provided with a real-life anonymized data set and are given the opportunity to analyze it and provide insight into the underlying business process. Competition entries are submitted in the form of a paper manuscript, and a winner is selected by a panel of judges.

2.2 Data

The Rabobank financial institution provided the data [7] for the 2014 challenge. This particular data was generated from its ICT operations, where their approach to responding to calls and technical support issues is rooted in *change management*. Thus when Rabobank discovers that a higher than usual number of calls are being reported for a specific service component, a *change* is scheduled for the offending service component(s), which is expected to rectify the underlying issue. Changes are recorded in the database and linked to the service component in question.

In contrast to previous years where data was often given in one large event log, data in the 2014 challenge was given in four separate tables: A table of *interactions* with calls to the service desk, a table of *incidents* which contains records of those interactions that could not be resolved by the first contact and thus had to be assigned to a service team, a table of *incident activities*, which simply contains a log of those activities performed by the service team(s) to resolve the particular incident, and a table of *changes*.

2.3 Challenge Questions

Rabobank posed the following questions, as taken from [8]:

Identification of Impact-patterns: We expect there to be a correlation between the implementation of a change and the workload in the Service Desk (SD) and/or IT Operations (ITO), i.e. increased/decreased volume of Closed Interactions and/or increased/decreased volume of Closed Incidents. Rabobank Group ICT is interested in identifying any patterns that may be visible in the log for various service components to which a configuration item is related, in order to predict the workload at the SD and/or ITO after future changes.

Parameters for every Impact-pattern: In order to be able to use the results of prior changes to predict the workload for the Service Desk directly after the implementation of future changes, we are interested in the following parameters for every impact-pattern investigated in sub question 1:

1. What is the average period to return to a steady state?
2. What is the average increase/decrease of Closed Interactions once a new steady state is reached?

Change in Average Steps to Resolution: Since project managers are expected to deliver the same or better service levels after each change implementation, Rabobank Group ICT is looking for confirmation that this challenge is indeed being met for all or many Service Components.

Creativity challenge: Finally, we challenge the creative minds, to surprise Rabobank Group ICT with new insights on the provided data to help change implementation teams to continuously improve their Standard Operation Procedures.

3 High Level Holistic View

3.1 Data Preprocessing

3.1.1 Restricting the data to the Rabobank selection criteria.

The original Rabobank data was contained in 4 comma-separated values files. These files were read by a computer program in order to integrate all records in a common event structure holding references to a event unique identifier, open time, close time, case identifier (configuration item name), and all other data fields and values in a list of associated feature key-value pairs. In the case of incident activities records, the missing configuration name and configuration type information was recovered by looking up those values in either the interaction or incident events referred to in the incident activity record.

	Changes	Interactions	Incidents	Incident Activities	Events	Cases (CI)
All data	30,273	147,004	46,606	466,737	690,620	14,144
In time window	25,976	145,478	45,456	449,568	666,478	12,327

Table 1: Number of data records.

	Changes	Interactions	Incidents	Incident Activities
Open Date	Change.record Open.Time	Open.Time First.Touch	Open.Time	DateStamp
Closed Date	Change.record Close.Time	Close.Time	Close.Time	DateStamp

Table 2: Selected fields for open and close dates

The data used for the analysis was then reduced to the time window of October 1st 2013 to March 31st 2014. This decision was made to avoid having events in the analysis that either were initiated before October 1st 2013, or that were not closed by March 31st 2014. So all event records in the analysis had to have both their open and close dates within that date interval. Table 1 summarizes the number of records from the original data, those fitting the time window for each event types, and their indexation using the configuration item names. Table 2 indicates what original data fields were used to assign the open and close times for each event types. All the event time stamps used the close time values in the holistic analysis.

3.1.2 Large item sets analysis.

Given that the challenge questions were mostly focused on the impact-pattern related to various service components related to the workload of the workload of the Service Desk and/or IT Operations, we selected a subset of features to reduce the number of feature combinations. For all event types, we used the configuration item name type (CI.Type.aff) and subtype (CI.Subtype.aff) as a means to look at the various service components. In addition, we included the category field (Category), thinking that because of its values (Request for information, or Incident), it could be a good indicator of the impact of a change from interactions and incidents following that change.

The a priori algorithm was applied to the challenge data in an incremental manner to identify the large item sets meeting a frequency minimum support. The item sets were defined as sets of feature key-value pairs. The analysis started at 1% support with increments of 1%. The total number of cases (CI.Name.aff) was 12,327, indexing 666,478 events.

Table 3 presents the number of large item sets that were generated to meet the 1% baseline minimum support. The algorithm reached the largest item sets for the minimum support baseline after 10 iterations. Then 7 additional

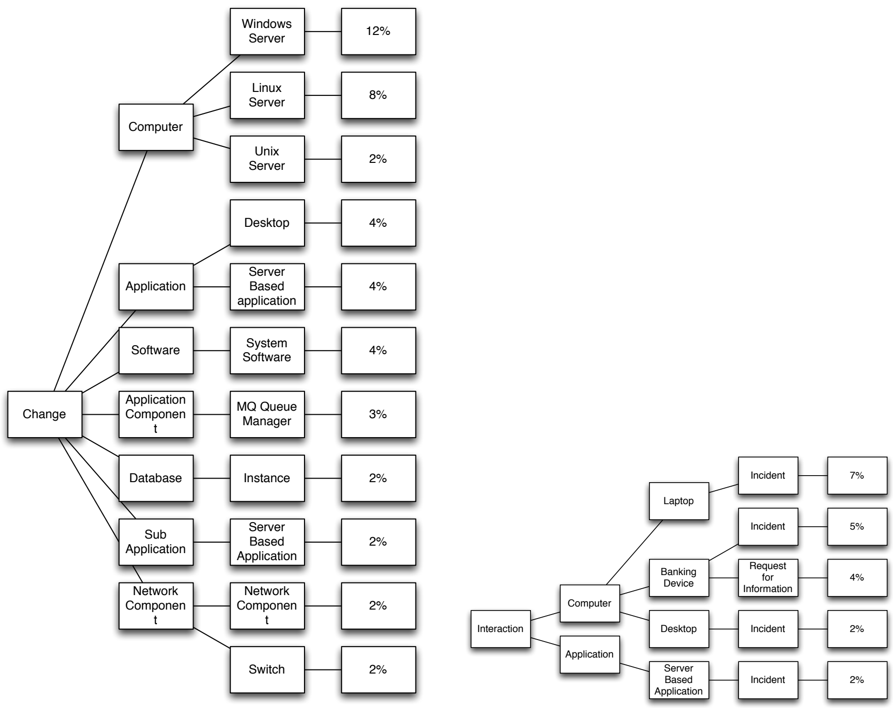

Figure 1: Hierarchical decomposition of large item sets containing a change (left) and interaction (right) event types

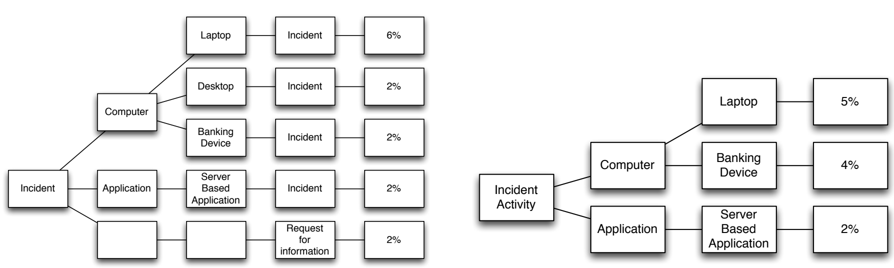

Figure 2: Hierarchical decomposition of large item sets containing an incident (left) and an incident activity (right) event type

Iteration	Nb of sets	Min support	Set composition (set-size count)
1	36	1%	((1 36))
2	85	1%	((2 85))
3	54	1%	((2 6) (3 48))
4	38	1%	((2 6) (3 21) (4 11))
5	38	1%	((2 6) (3 21) (4 11))
6	24	2%	((2 1) (3 14) (4 9))
7	13	3%	((3 8) (4 5))
8	12	4%	((3 7) (4 5))
9	6	5%	((3 3) (4 3))
10	4	6%	((3 2) (4 2))
11	4	7%	((3 2) (4 2))
12	1	8%	((3 1))
17	0	13%	()

Table 3: Incremental minimum support analysis using the apriori algorithm

iterations were necessary to reach a point were no item sets met the minimum support (13%).

Then a hierarchical decomposition of the item sets was done to identify the features to use in further analysis. The Figures 1, and 2 show the item set in a hierarchical manner from the event types, to the configuration type, subtype, and category. The leafs in the figures give the maximal support obtained by each item set. An inspection of the item sets indicated that a 4% minimum support would meet the criteria that the feature sets would contain the 4 main event types (change, interaction, incident, and incident activity), but 2% would cover more services, and the category field. However, the hierarchical decomposition of item sets also shows a lack of variability in the crossing of services, and interaction and incident category (Request for information or incident). Also the most frequent configuration types (Application and Computer) appear to be distributed over all event types (interaction, incident, change, and incident activities). Therefore, the remaining analysis will proceed to answer the challenge questions using the application and computer services, as well as the four event types, but focusing mainly on closed interactions and incidents.

3.2 The challenge questions

In order to answer the first challenge question, a set of common data analysis procedures were use to answer all questions. Unless it is specified otherwise in the data analysis section related to a question, these common procedures have been applied.

3.2.1 Common data processing procedures: first closed change event.

The analysis counted the sum of closed interactions and incidents occurring before and after the first closure of a change event in the series of events related

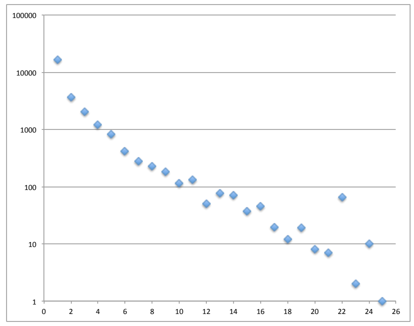

Figure 3: Frequency of change events as a function of their durations in weeks

to a configuration item. We assumed that the analysis of the events following the earliest change closure could indicate the patterns relevant to answer the challenge questions. The first closed change event is not necessary the first event in the series; in this case the open time of the event would be used. Even though, most change events have a short duration between their opening and closure, it is possible that a change event gets closed before another one that had been initiated before. The Figure 3 shows the frequency of change event durations.

3.2.2 Common data processing procedures: time duration between events.

The time difference between the first closed change, and other events were rounded up or down to the upcoming, or preceding 7 days. For example, given that the closed change will be on day 0, all events occurring during the first week will be counted as if they happened on day 7, and so on. The same calculation was applied to events preceding the closed change event. The Figure 4 presents the sum of interaction, incident, and change events in a event sequence containing a closed change. The incident activities were left out of the figure because of their high frequencies, which hides the distribution patterns of the other events. Also to notice that there are no change events before time 0. This is the result of using only closed times as event time stamps. The figure also shows a dip in the distribution of the frequencies, this pattern is very frequent in other figures and happens to correspond to the Christmas Holidays.

3.2.3 Question 1: Identification of Impact-patterns.

We expect there to be a correlation between the implementation of a change and the workload in the Service Desk (SD) and/or IT Operations (ITO), i.e. increased/decreased volume of Closed Interactions and/or increased/decreased volume of Closed Incidents. Rabobank Group ICT is interested in identifying

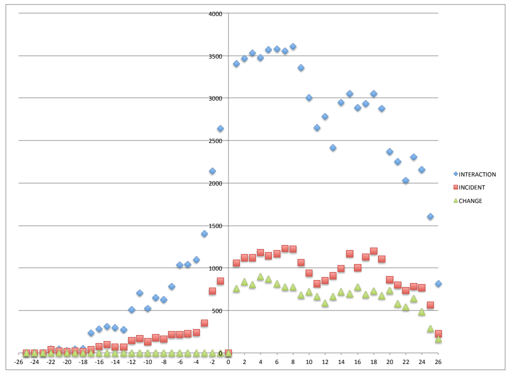

Figure 4: Sum of interaction, incident, and change events as a function of the time differences to the first closed change

any patterns that may be visible in the log for various service components to which a configuration item is related, in order to predict the workload at the SD and/or ITO after future changes.

3.2.4 Data analysis.

The common data processing procedures were applied to produce the event frequency distributions. As outlined in the large item set analysis, we focused the analysis by comparing the application and computer services in terms of the distribution of closed interactions and incidents. Figure 5 presents the sum of data records for each services, while Figure 6 show respectively the correlation between interaction frequencies, and incident frequencies.

3.2.5 Discussion/Answer to the question.

The distribution of closed interactions and incidents events shows clearly a rapid increase of interactions and to a lesser extend incidents, preceding the closure of a change event. This pattern is similar when inspecting all services or the specific application and computer service components. However, the distributions seems to have different patterns when comparing the application to the computer services (Figure 5). The application service component shows a rapid increase followed by a constant decrease. On the other hand, the computer service indicates a constant increase during the 26 weeks period followed by a rapid drop at the end of the period. Figure 6 show an R-squared correlation between services interactions (0.219) and incidents (0.119). This is in sharp constrast with the R-square values for the within service correlations between interactions and incidents (Application=0.900, Computer=0.677). Therefore, the data gives some support to the distinction of patterns between the application and

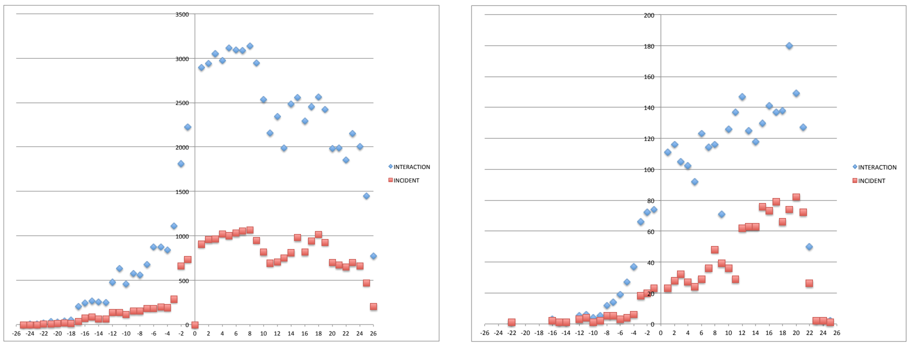

Figure 5: Sum of closed interaction, and incident events for the application (left) and computer (right) services as a function of the time differences to the first closed change

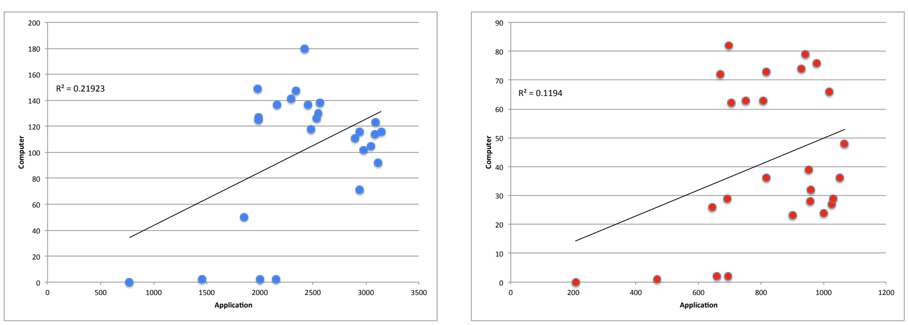

Figure 6: Correlation between interaction (left) and incident (right) frequencies for the application and computer services

computer services both in terms of the distribution of closed interactions, and closed incidents.

3.2.6 Question 2: Parameters for every Impact-pattern.

In order to be able to use the results of prior changes to predict the workload for the Service Desk directly after the implementation of future changes, we are interested in the following parameters for every impact-pattern investigated in sub question 1:

1. *What is the average period to return to a steady state?*
2. *What is the average increase/decrease of Closed Interactions once a new steady state is reached?*

3.2.7 Data analysis: steady state

A possible interpretation of what a steady state would be when no change is being processed within the scope of a configuration item. The series of events in this situation would reflect a flow of interactions and incidents with no associated change. Figure 7 show the respective sum of events for those series for a) all services, b) the application service, and c) computer service.

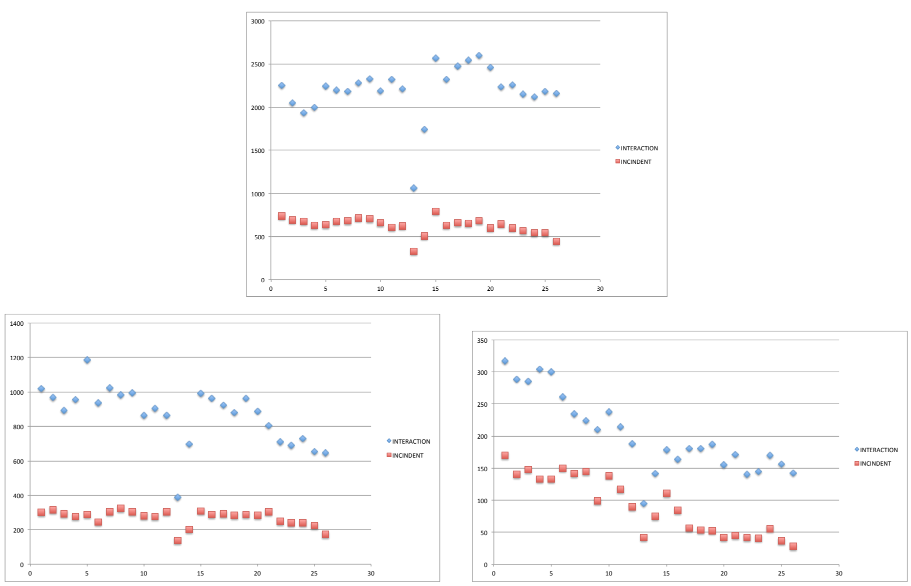

Figure 7: Steady state (event series with no change) frequency distribution for all services (top), and the application (bottom left) and computer (bottom right) services.

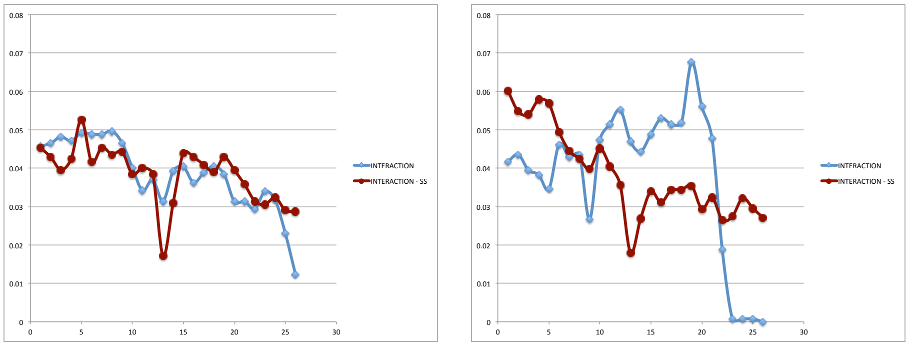

Figure 8: Comparison of interactions and steady state interactions for the application (left R-square=0.509) and computer (right R-square=0.041) services

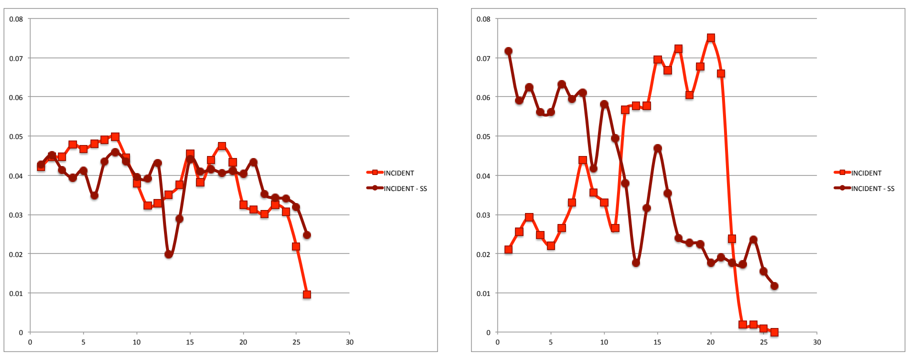

Figure 9: Comparison of incidents and steady state incidents for the application (left R-square=0.358) and computer (right R-square=0.018) services

3.2.8 Data analysis.

In order to answer the second question, we looked at the distribution of steady states in relation to the distributions of closed interactions and incidents when a closed change occurs. Figure 8 compares the interactions and steady state interactions for the application and computer services, while Figure 9, does the same for the distribution of incidents.

3.2.9 Discussion/Answer to the question.

The inspection of the distribution of frequencies in the figures 8 and 9 suggests that aside from the data for all services (Figure 7), the steady state for the 26 weeks shows for both services a decline in the number of interactions and incidents over time. This pattern seems to reflect the reality that when a configuration item does not require a change, then one would expect that early interactions and incidents should clarify the issue with a tendency to reduce the number of events. So instead of looking at an absolute value for a number of events as a measure of a steady state, it seems more appropriate to compare the distribution of steady state events to the ones that are related to a change. Essentially, a distribution of events located over the steady state line would indicate an increase of events over what is expected from a steady state. A distribution equal or below the steady state line, on the contrary would indicate that the events generated by a closed change do not increase the related events over what is expected from a steady state. In this respect, only events over the line of the steady state should attract attention in terms of measuring the average period to reach a steady state.

From a visual inspection and from R-squared values, it seems that only the computer service should be of concern as to when it returns to a steady state. So the answer to the sub-question 1 is, for closed interactions, the computer service leaves the steady state at week 10 and returns at week 22. This would mean 12 weeks for interactions to return to a steady state. The situation is similar for incidents, leaving the steady state at week 11 to return at week 22. The answer to the sub-question 2 is that for all services the pattern is a sharp decline of interactions and incidents after coming back to a steady state.

3.3 Question 3: Change in Average Steps to Resolution.

Change in Average Steps to Resolution: Since project managers are expected to deliver the same or better service levels after each change implementation, Rabobank Group ICT is looking for confirmation that this challenge is indeed being met for all or many Service Components.

3.3.1 Data analysis.

Similar to the previous question where the notion of a steady state was left to interpretation, the reference to levels of service was not provided. We decided to measure the number of incidents (from the field NB.Related.Incidents) occuring for each closed changes within the same configuration items as a measure of service level. The reasoning being that if managers deliver better (or worst) services, the number of related incidents should go up (or down) over time. Figure 10 shows the frequency distribution of those incidents as a function of when a closed item occurred during the 26 weeks period (right) and the comparison of incidents and steady state incidents for all events.

3.3.2 Discussion/Answer to the question.

Figure 10 (left) suggests an even distribution over time of incidents events related to changes. Even with the two data points that are higher than the other ones, one might conclude that there is no improvement or deterioration of management intervention effects over time. Figure 10 (right) is congruent with this

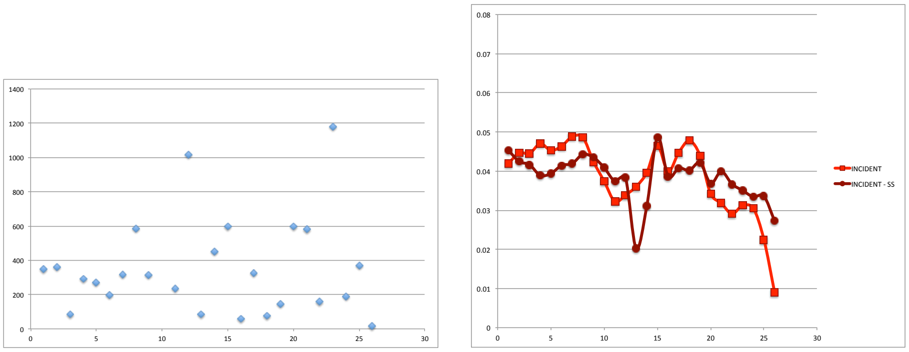

Figure 10: Distribution of related incidents to a change as a function of time (left) and comparison incidents and steady state incidents for all events (right).

interpretation because the incidents and steady state incidents do not indicate any major difference in their patterns, pointing to good management practice.

4 Low Level Analysis

Before conducting the low-level analysis and addressing the challenge questions, some background on sequence mining is provided.

4.1 Sequence Mining

Throughout this section, consider I to be a set of items and T to be a set of transactions containing itemsets from I. Also, for the sake of illustration, consider a running example where $I = \{a, b, c, d, e, f, g, h\}$ and $T = \{\{a, b\}, \{c, d, e\}, \{c, e, f\}, \{a\}, \{d, f\}, \{c\}, \{e, g, h\}, \{f, g, h\}\}$.

4.1.1 Association Rule Mining

The goal of association rule mining [1, 10] is to find items that appear together with sufficient frequency in itemsets in T. The frequency in which items appear together is represented in terms of *support*. The support $supp(i)$ of an itemset i is equal to $|\{t \in T | i \subseteq t\}|/|T|$, or the proportion of transactions that contain i. An association rule $X \rightarrow Y$ for itemsets X and Y with $X \cap Y = \phi$ indicates that, whenever any transaction t contains the items in X, t will also contain Y. The support of $X \rightarrow Y$ reflects the number of positive instances of the rule, and is given by $supp(X \cup Y)$, while the *confidence* of $X \rightarrow Y$ indicates the percentage of transactions containing X for which the rule is true, and thus Y is also present, and is computed by $supp(X \cup Y)/supp(X)$. It is typically the goal in association rule mining to identify such rules that meet or exceed some prespecified minimum thresholds for rule support and confidence.

Consider the values given for I and T in the running example, as well as the association rule $\{c\} \to \{e\}$. The support of $\{c\} \to \{e\}$ is $supp(\{c, e\}) = 0.25$, while the confidence of $\{c\} \to \{e\}$ is $supp(\{c, e\})/supp(\{c\}) = 0.67$

4.1.2 Sequential Pattern Mining

While association rule mining aims to discover *intra*-transactional patterns, sequential pattern mining [2, 6] aims to discover *inter*-transactional patterns. Consider the above model, with the additions that each transaction has (1) a *time* attribute, which produces a total order over T, and (2) a *case* attribute, which gives a partition over T. As a result, T becomes a set S of ordered lists of itemsets, referred to as *sequences*, which are henceforth written as $\langle t_1 t_2 \ldots t_n \rangle$.

A sequence $\langle a_1 a_2 \ldots a_n \rangle$ is *contained in* another sequence $\langle b_1 b_2 \ldots b_m \rangle$ if there exist integers $i_1, i_2, \ldots, i_n$ such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \ldots, a_n \subseteq b_{i_n}$. A sequence $s \in S$ *supports* a sequence s' if s' is contained in s. The *support for a sequence* s' is then $|\{s \in S | s \text{ supports } s'\}|/|S|$, which is the fraction of sequences in S that support s'. The goal of sequential pattern mining is to find the set of *maximal* sequences whose support meets or exceeds some prespecified minimum threshold, where a sequence is maximal in a set if it is not contained in any other member in the set.

Referring back to the example, suppose transactions in the set T are temporally ordered and divided into three cases, resulting in the sequences $S = \{\langle\{a, b\}\{c, d, e\}\{c, e, f\}\rangle, \langle\{a\}\{d, f\}\{c\}\rangle, \langle\{e, g, h\}\{f, g, h\}\rangle\}$. The sequence $\langle\{a\}\{d\}\{c\}\rangle$ is supported by both the first and second sequences, and thus has support $2/3$. Given a minimum threshold of 0.6, sequences $\langle\{e\}\{f\}\rangle$ (support $= 2/3$) and $\langle\{f\}\rangle$ (support $= 1$) meet the minimum threshold, however $\langle\{f\}\rangle$ is not deemed maximal as it is contained in $\langle\{e\}\{f\}\rangle$.

4.1.3 Sequence Classification

Sequence classification [4, 5, 9] is the field of study that attempts to classify sequences in S by using frequent sequential patters as features in the classification. Consider the above model with the addition of a set L of *class labels*, where each $s \in S$ is labeled with an element of L. S is now a set of *examples*, where each example $s \in S$ can be represented by a set of feature-value pairs using features from the set S' of frequent sequential patterns and boolean values. A feature f thus holds the value "true" if s contains f, and "false" otherwise. For example, consider features $f_1 = \langle\{a\}\{d\}\{c\}\rangle, f_2 = \langle\{e\}\{f\}\rangle, f_3 = \langle\{e\}\{g\}\rangle$. The sequence $\langle\{a, b\}\{c, d, e\}\{c, e, f\}\rangle$ could then be represented by the feature-value pairs $(f_1, true), (f_2, true), (f_3, false)$.

The goal of sequence classification is to identify sequences for the feature set that have the following properties:

- Features should be frequent
- Features should be distinctive of at least one class

- Feature sets should not contain redundant features

For a more detailed example, consider the feature set $F = \{\langle\{a, b\}\{c, d, e\}\{c, e, f\}\rangle, \langle\{a\}\{d, f\}\{c\}\rangle, \langle\{e, g, h\}\{f, g, h\}\rangle\}$. Each sequence in S can be described in terms of boolean feature-value pairs as follows:

$\langle\{a\}\{d\}\rangle$	$\langle\{e\}\{f\}\rangle$	$\langle\{g, h\}\{g, h\}\rangle$	$\langle\{a\}\{f\}\rangle$	Sequence	Label
1	1	0	1	$\langle\{a, b\}\{c, d, e\}\{c, e, f\}\rangle$	c_1
1	0	0	1	$\langle\{a\}\{d, f\}\{c\}\rangle$	c_2
0	1	1	0	$\langle\{e, g, h\}\{f, g, h\}\rangle$	c_1

If the sequences in S are labeled as in the above graphic (i.e. s_1 and s_3 as c_1 and s_2 as c_2), then the feature $\langle\{e\}\{f\}\rangle$ appears to conform to these desired properties, and as a result plays a dominant role in distinguishing between the two classes (true implies c_1, false implies c_2).

The typical methodology for sequence classification involves two phases: (1) sequential pattern mining to discover the frequent sequences that potentially will make good feature candidates, and (2) some sort of assessment of features (e.g. using an "interestingness" measure) to select the best candidates. From there, standard statistical classification methods are used to label future instances. The key problem that is faced in this domain of research is the vast space of potential features. Most research efforts appear to be centered on mitigating this problem.

4.2 Analysis

4.2.1 Linking the Data

While each record in the Incidents table relates to a particular interaction, and each record in the Incident Activities table relates to a particular incident, there is no explicit relation of incidents and interactions to changes, other than a very small number (536/46606) of incident records that are known to have been caused by a particular change. In this case, the incidents refer directly to the changes from which they are proven to have resulted. Thus any effort to measure an increase/decrease in ICT operations workload after a change is implemented cannot rely on any specific reference between changes and interactions, incidents and activities. Moreover, neither can any attempt to discover patterns in relevant interactions, incidents and activities before the change is implemented that might be used to predict workload after the change.

Instead, each record in every table (interactions, incidents and changes explicitly, incident activities via their relevant incidents) refers to a "Configuration Item" (CI), which is linked to a specific service component. Thus one needs to examine the events that are relevant to the CI affected by a change in order to

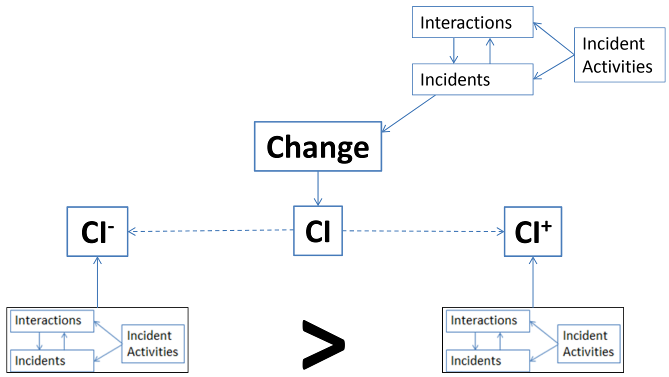

Figure 11: Links amongst the four tables, centered around a change

find predictive patterns of activities that occur before a change, and to measure any changes in workload that result after a change.

Figure 11 depicts the links available in the various tables, centered around a particular change. The events in the upper right represent the small number of consequential events that proven to be caused by the change, while the events in the lower left and right represent the events relevant to the CI before the change occurs (shown as CI-) and the CI after the change occurs (shown as CI+), respectively. As further explained in more detail in the next section, the expectation is that the number of events associated with CI+ should be lower than that associated with CI-.

While each interaction is associated with an affected CI, each incident and, by association, each incident activity, is associated with both an affected CI and a "caused-by" CI, meaning the CI that actually caused the problem as opposed to the CI that was affected by the problem. Since each incident points to an interaction, the caused-by CI can thus be determined for many interactions. However almost 2/3 of interactions are resolved at the service desk and have no associated incident, meaning that a majority of interactions are only associated with an affected CI. The BPI 2014 "Quick Reference" guide specifies that a change may be scheduled "if particular service disruptions reoccur more often than usual" [8], and that this change occurs on the caused-by CI. Thus the expectation is that there should be a rise in the number of relevant events before a change (leading up to the "Actual Start" time), and a decrease afterwards (following the "Actual End" time). Figure 12 verifies that this pattern does indeed occur with the number of incidents caused by the CI, with trend lines indicating that the number of incidents increases by about 11% over the 20-day period before a change, and decreases by about 25% after a change.

The question is whether this pattern is apparent in the number of interactions, considering that we only know the caused-by CI for roughly 1/3 of

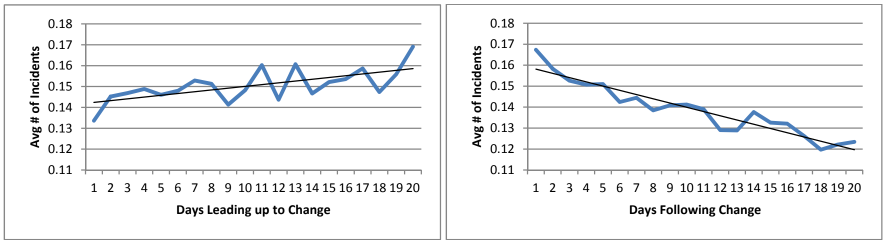

Figure 12: Average number of incidents per change, caused by the change's affected CI, for 20 days preceding the actual start of a change and 20 days following the actual end

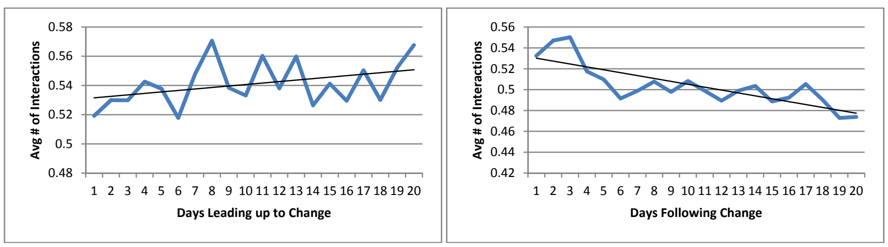

Figure 13: Average number of interactions per change, where affected CI matches change's affected CI when caused-by CI is not available, for 20 days preceding the actual start of a change and 20 days following the actual end

them, and must then match the affected CI with the change's affected CI to find the relevant interactions amongst the other 2/3. Figure 13 shows that the expected pattern is still present, but not as pronounced, with trend lines indicating approximately a 4% increase and a 9% decrease before and after a change, respectively.

4.2.2 Average Time to Steady State and Subsequent Increase/Decrease in Workload

The key goal in the change management model is to implement changes that will improve the level of performance or success of the business unit. For Rabobank ICT, the key is to improve the reliability of the technical services that they oversee, which can be measured in terms of the workload experienced at the service desk and or IT operations. Thus the expectation is that a change implemented for a particular CI, perhaps after an initial spike in service calls, should eventually result in a reduced workload for that CI. Daily workload after implementation of a change might thus follow something similar to the left curve in Figure 14. Simple inspection of this particular graph might lead one to believe

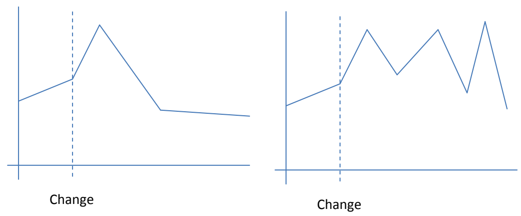

Figure 14: Typical curve plotting workload before and after a change without noise from other changes (left) and with noise from other changes (right)

that the steady state is reached where the line begins to flatten. In reality, however, determining the time to steady state is not quite so simple, due to the fact that multiple changes might be occurring continuously for a particular CI, creating noise. In that case, one might find a curve resembling the right of Figure 14. Since there are multiple spikes after the change, it is difficult to know exactly what portion of the daily workload is attributable to the change in question, and thus where the steady state is reached.

Since the average time to steady state thus cannot be found by examining workload around individual CI, we attempted an approach that aggregates the results and finds correlations between daily workload and *the number of recent changes*. The hypothesis was that the higher the number of recent changes that were implemented, the higher the workload should be. The approach was to vary the size of the window in which to sum the number of changes, and find which size window gives a daily number of recent changes that correlates maximally with workload

Figure 15 depicts the number of incidents opened each day during Oct 1/13-Mar 31/14, as well as the number of changes implemented each day (according to the "Actual End" date), for CI SBA000263. The goal is thus to locate the spike in workload attributable to each change. Manual inspection of this particular graph is likely to be fruitless. However, as we expand the window in which we sum the most recent changes, for example by graphing the number of changes occurring in the previous 10 days (as depicted in Figure 16), a correlation with the number of incidents starts to emerge.

Figure 17 depicts the correlation between the number of daily incidents and the number of changes in each day, the number of changes the last two days, three days, and so on, up to twenty. The second line (held constant at 0.195) shows the minimum level required to achieve statistical significance at the $p = 0.05$ level given the number of points. Thus this indicates that there is a significant positive correlation between the number of incidents in a day and the number of changes made in the previous 9 days, as well as the number of changes made in the previous 10 days, all the way up to 20. Thus we can conclude that there is a relationship between a change and a subsequent rise in workload, for CI SBA000263.

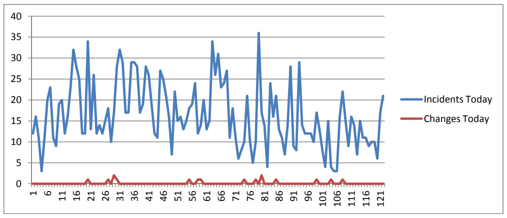

Figure 15: Number of incidents and changes each day during Oct 1/13-Mar 31/14 for configuration item SBA00263

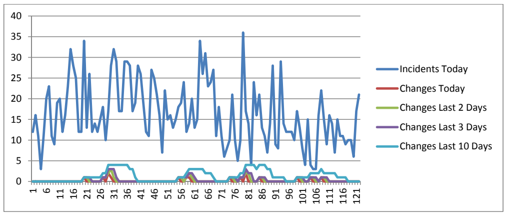

Figure 16: Number of incidents each day and number of changes in previous 1, 2, 3 and 10 days during Oct 1/13-Mar 31/14 for configuration item SBA00263

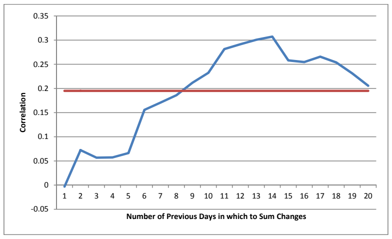

Figure 17: Correlation between the number of daily incidents and the number of recent changes, from the previous 1 day to the previous 20 days

More interestingly, this graph also contains information that can be used to estimate the average time to steady state. Referring back to Figure 17, we see that there is basically 0 correlation between the number of changes in a day and the number of incidents. However, looking at the number for changes in the previous two days increases the correlation to about 0.07, which means that there is more of a relationship between the number of changes in the last two days and the number of incidents. This increase generally continues until the peak is reached at 14, finally decreasing for good at 15 days. Since the correlation between the number of incidents in a day and the number of changes in the previous 15 days is lower than that in the previous 14 days, this implies that considering any changes that occurred 15 days ago will not add to the correlation, and thus there must be no (significant) relationship between the number of incidents in a particular day and any changes that occurred 15 days prior. This implies that the new steady state is reached.

We computed the correlation between the number of incidents and the number of changes using windows from 1 to 20 days for the 11 configuration items that caused at least 250 incidents and were affected by at least 10 changes opened and closed during the Oct 1/13-Mar 31/14 period. Figure 18 depicts those graphs for the remaining 10 configuration items. Table 4 then indicates the average number of days to steady state for each CI (where significant positive correlation occurs) using the approach outlined above.

To answer the question of average increase/decrease in the number of closed interactions, we used the average time to steady state for each of the nine configuration items mentioned in the previous section that experienced a significant correlation between the number of changes and workload, and computed:

1. The average increase in the number of daily interactions experienced during the spike in workload after a change (i.e. the period between the "Actual End" of the change and the return to steady state

2. The average decrease in the number of daily interactions in the steady state after the change

Each of these was measured against the average daily interactions experienced in the 10 days before the "Actual Start" time of the change. In the case of the average decrease measurement, the daily interactions in the 5 days after the return to steady state were used. Figures for each of the 9 configuration items that experienced significant correlation are given in Table 4.

4.3 Identifying Predictive Patterns

The final step is to identify any patterns in the data occurring before a change that may help predict workload after a change, both during the immediately following spike as well as in the new steady state. To accomplish this, we employed the use of sequence classification on ordered lists of events that appear before a given change is initiated. In the case of trying to predict whether there

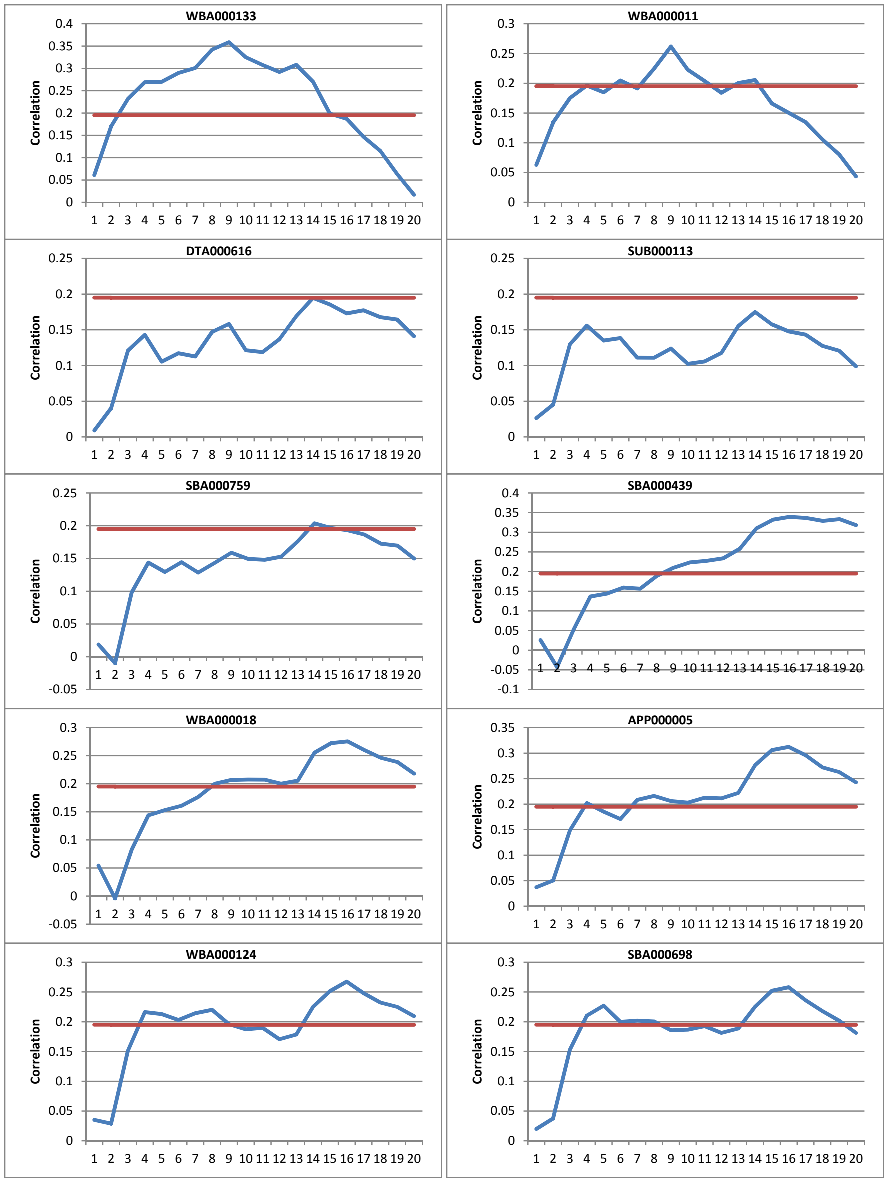

Figure 18: Correlation between the number of daily incidents and the number of recent changes, from the previous 1 day to the previous 20 days, for all CI (other than SBA000263, shown in Figure 17) that caused at least 250 incidents and were affected by at least 10 changes

CI	Days to Steady State	Avg Before Change	Avg After Change before Steady State	Avg Steady State	% Decrease
SBA000263	12	27.4	25.5	21.0	23%
WBA000133	10	58.7	56.5	55.4	6%
WBA000011	10	16.9	18.4	18.1	-7%
DTA000616	n/a	n/a	n/a	n/a	n/a
SUB000113	n/a	n/a	n/a	n/a	n/a
SBA000759	15	4.9	4.9	4.6	6%
SBA000439	17	44.5	40.4	37.4	16%
WBA000018	17	10.0	6.2	6.7	33%
APP000005	17	19.3	18.9	18.3	5%
WBA000124	17	7.9	8.4	8.3	-5%
SBA000698	17	6.5	3.7	2.9	55%
Average	15	25.5	24.0	22.7	11%

Table 4: Average number of days to steady state

would be a decrease in workload (in this case, defined as the number of interactions opened for the change's affected CI) after the steady state is reached, the set of changes (affecting the same 9 configuration items studied above) were partitioned according to whether or not the steady state workload following the change experienced a larger-than-average decrease. For each change, a sequence of "lead-up" events was mined, consisting of the incidents that were caused by the change's CI in 5 days prior to the actual start of the change. Each incident was represented by its "Closure Code" in the sequence. To consider the effect of having multiple recent changes, we also included any other changes that were implemented in the prior 5 days, where each change was represented by its "Change Type". Thus there were two sets, or "classes" of sequences: those associated with changes that led to higher than average decreases in workload, and those associated with changes that did not. Sequence classification was then performed to mine the representative susbsequences from each class that could be used to classify future "lead-up" sequences, which could then be used to predict whether the change being considered would likely result in decreased workload at the service desk.

Similarly, changes were also partitioned as to whether or not an immediate increase, or a "spike", in workload was experienced in the first 3 days following the implementation of a change. Sequence mining was again used to find patterns in the lead-up events that could be used to predict whether future changes are likely to cause short, immediate increases in workload.

We employed a naive Bayes classifier that mined 10 subsequences as features for each class. Representative subsequences where confined to include no more than 4 events. The results of the classification for the first test, predicting higher-than-average decrease in workload at steady state, are given in the first row of Table 4.3. The 10 selected features for each class, as well as their pre-

Prediction	True Pct	Precision	Recall
High decrease in steady state	51%	60%	98%
High dec in steady state no changes	51%	60%	98%
Immediate increase after change	55%	61%	95%

Table 5: Performance of three predictive sequence classifiers

Sequence	PPV	NPV
Standard Change Type 06, No error - works as designed, Data, Other	0.941	0.525
Standard Change Type 06, Other, Software, Data	0.933	0.520
Other, Standard Change Type 06, Data, Software	0.929	0.518
Other, Standard Change Type 06, User manual not used, Other	0.929	0.518
Software, User manual not used, Standard Change Type 06	0.929	0.518
Software, Standard Change Type 06, User manual not used, Other	0.929	0.518
No error - works as designed, Inquiry, Other, Other	0.929	0.518
Standard Change Type 06, User manual not used, Other	0.929	0.518
Standard Change Type 06, Standard Change Type 06, Other, Other	0.929	0.518
Standard Change Type 06, Standard Change Type 06, Software, Other	0.929	0.518

Table 6: Sequences most predictive of higher-than-average decrease in workload after change at steady state

dictive power in terms of positive predicted value (PPV), which measures how well the sequence predicts the class it represents, and negative predictive value (NPV), which measures how well the sequence does not predict the wrong class, are given in Tables 4.3 and 4.3.

Two interesting observations can be made here. First note how sequences indicative of low decrease tend to include incidents coded as "User error", "Operator error", "No error", etc. This makes sense since, if there is really no error to fix, changes are not as likely to help bring down the number of future calls than, say, a user awareness/education program. The second interesting piece of information is that sequences indicative of high decrease tend to include changes of type "Standard Change Type 06". This gives the indication then that changes occurring soon after this type of change tend to be more effective. To complement this finding and further study the effectiveness of including previous changes, we ran the same test using only incidents in the lead-up sequences. The second row of Table 4.3 shows the results are exactly the same with or without changes. Thus, while we found that the presence of "Standard Change Type 06" changes indicates that future changes will be more effective, in general, including changes did not help prediction. Tables 4.3 and 4.3 give the most predictive sequences when changes are not included. In this case, more sequences that predict high decrease tend to include software-based incidents, which are more likely to offer the potential to improve operation and thus reduce

Sequence	PPV	NPV
Software, Operator error, User error, Other	1.000	0.557
Unknown, Software, User error, No error - works as designed	1.000	0.557
Operator error, User error, Other	0.913	0.553
Other, Operator error, User error, Other	1.000	0.778
Software, Operator error, User error, No error - works as designed	1.000	0.555
User error, Unknown, Software, No error - works as designed	1.000	0.555
Unknown, Other, User error, No error - works as designed	1.000	0.555
Operator error, User error, No error - works as designed	0.952	0.553
Other, User error, No error - works as designed, User error	0.552	0.553
Software, Unknown, User error, No error - works as designed	0.552	0.553

Table 7: Sequences most predictive of lower-than-average decrease in workload after change at steady states

future problems.

The third row gives results of classification of sequences that predict an increase in workload in the three days immediately after a change is implemented. Tables 4.3 and 4.3 give the sequences that predict increase and those that predict no increase, respectively.

5 Conclusion

This paper presented results and findings from our research conducted for the 2014 Business Process Intelligence (BPI) Challenge, using data sets provided by Rabobank's ICT operations. We tackled Rabobank's proposed challenge quesions using two approaches: a macro approach where conclusions were drawn via analysis of aggregated data from the entire set, and a micro approach where sequence classification was used to uncover predictive patterns in the data for individual configuration items. Each approach addressed the challenge questions from different levels, and offered insight on the findings from each point of view that can be applied to improve company business processes and decision making in the future.

References

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

Sequence	PPV	NPV
Software, Other, Other, Questions	0.917	0.513
Software, Other, Questions, Other	0.909	0.511
Questions, Other, No error - works as designed	0.909	0.511
Questions, Other, Other, No error - works as designed	1.000	0.513
Other, Software, Other, Questions	0.900	0.509
Data, Software, Operator error, User manual not used	0.900	0.509
No error - works as designed, No error - works as designed, No error - works as designed, Unknown	0.875	0.518
Other, Data, Operator error, User manual not used	0.889	0.506
Other, Questions, Other, No error - works as designed	0.889	0.506
Software, Unknown, Unknown, Operator error	0.889	0.506

Table 8: Sequences without changes that are most predictive of higher-than-average decrease in workload after change at steady states

Sequence	PPV	NPV
Software, Unknown, User error, No error - works as designed	1.000	0.560
Software, Operator error, User error, Other	0.955	0.555
Unknown, Other, User error, No error - works as designed	1.000	0.557
Unknown, Software, User error, No error - works as designed	1.000	0.557
Other, Operator error, User error, Other	1.000	0.555
Software, Operator error, User error, No error - works as designed	1.000	0.555
User manual not used, User error, No error - works as designed, Software	0.952	0.552
Operator error, User error, No error - works as designed	0.952	0.552
Unknown, Other, No error - works as designed, User error	0.950	0.550
Operator error, User error, Other, Other	0.950	0.550

Table 9: Sequences without changes that are most predictive of lower-than-average decrease in workload after change at steady states

Sequence	PPV	NPV
No error - works as designed, User manual not used, User manual not used, Software	0.929	0.564
No error - works as designed, Software, User manual not used, User manual not used	1.000	0.562
Questions, Other, No error - works as designed	0.909	0.559
Other, Software, Inquiry, Software	0.833	0.558
No error - works as designed, Software, Data, Operator error	0.833	0.558
Data, No error - works as designed, Software, User manual not used	0.833	0.558
Operator error, Data, Data, Other	0.833	0.558
User error, User manual not used, User manual not used, Software	0.833	0.565
Other, Questions, Other, No error - works as designed	1.000	0.559
Software, Inquiry, Other, Software	0.900	0.558

Table 10: Sequences that are most predictive of immediate increase in workload after a change is implemented

Sequence	PPV	NPV
Other, Operator error, User manual not used, No error - works as designed	0.944	0.476
Other, Operator error, User manual not used, Unknown	0.941	0.475
Other, Unknown, Operator error, User manual not used	1.000	0.474
User error, Software, Other, Operator error	0.933	0.472
Operator error, User manual not used, Other, No error - works as designed	0.895	0.475
No error - works as designed, User manual not used, User error, Unknown	0.929	0.471
Unknown, Operator error, User manual not used, Unknown	1.000	0.472
Data, No error - works as designed, Unknown, Data	0.929	0.471
Operator error, Other, No error - works as designed, User manual not used	1.000	0.472
Operator error, User manual not used, Unknown, Software	1.000	0.472

Table 11: Sequences that are most predictive of no immediate increase in workload after a change is implemented

[3] Fluxicon Blog. Bpi challenge 2014 ? get started now! http://fluxicon.com/blog/2014/04/bpi-challenge-2014. Date accessed: Jul 11, 2014, 2014.

[4] Neal Lesh, Mohammed J Zaki, and Mitsunori Ogihara. Mining features for sequence classification. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 342–346. ACM, 1999.

[5] Neal Lesh, Mohammed J Zaki, and M Oglhara. Scalable feature mining for sequential data. *Intelligent Systems and their Applications, IEEE*, 15(2):48–56, 2000.

[6] Carl H Mooney and John F Roddick. Sequential pattern mining–approaches and algorithms. *ACM Computing Surveys (CSUR)*, 45(2):19, 2013.

[7] Rabobank. Business processing intelligence challenge (bpic) data. DOI 10.4121/uuid:c3e5d162-0cfd-4bb0-bd82-af5268819c35, 2014.

[8] Boudewijn van Dongen, Diogo R. Ferreira, Barbara Weber, and Jochen De Weerdt. Business processing intelligence challenge (bpic). http://www.win.tue.nl/bpi/2014/challenge. Date accessed: Jul 11, 2014, 2014.

[9] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48, 2010.

[10] Qiankun Zhao and Sourav S Bhowmick. Association rule mining: A survey. *Nanyang Technological University, Singapore*, 2003.