Dotted Chart and Control-Flow Analysis for a Loan Application Process

Thomas Molka[1,2], Wasif Gilani[1] and Xiao-Jun Zeng[2]

Business Intelligence Practice, SAP Research, Belfast, UK
The University of Manchester, School of Computer Science, Manchester, UK
{thomas.molka, wasif.gilani}@sap.com, x.zeng@manchester.ac.uk

Abstract. In this paper we show our results of analysing a real-life event log taken from a Dutch financial institute. The log contains 262.000 events and describes a personal loan application process. We mine the control-flow model using the fuzzy miner from the process mining tool Disco, and analyse different variants of the process statistically and with dotted charts. This allows us to conclude performance properties of the historic execution of the process. Furthermore we examine the resources assigned to different process steps.

1 Introduction

In today's process supporting information systems more and more data is being generated and the need for process documentation is increasing. This is mainly for compliance reasons (SOX) but also for analysing and improving business processes in a quickly changing business environment. In most cases accurate process documentation is missing and even if there is documentation existing, it is not guaranteed that this information is up-to-date as certain properties of the process may change over time. In the area of Process Mining, methodologies have emerged that aim to provide insight into the behaviour of processes by mining formal models from event logs [1]. In addition to retrieving structural information about the control flow of the process, process mining techniques can also be used to get an organisational view on the process (resources working against process steps and relation between resources) and a perspective for analysing the performance of the process (e.g. finding bottlenecks) [2].

In this paper we report our results from the analysis of a process log provided for the Business Process Intelligence Challenge 2012 [8]. The log reflects a loan application process in a global financing organisation. The process starts with a customer applying for a loan, followed by some automatic checks and processing of the application by the resources working in the financial organisation. Examples of process steps include the submission of the application, adding information to it, calls between customer and financial organisation, declining, cancelling or accepting the application, etc.. The event log contains 262.200 events and 13.087 process instances (also: cases, traces). For each process instance the amount of the loan the customer is applying for is given. The events contain information such as a timestamp, the process step name, the resource processing the event and for some process steps also the lifecycle of the activity (scheduled, started or completed).

In the following sections we describe how we analysed the log using dotted chart analysis and control flow mining.

2 Analysing the log with dotted charts and statistics

To get an overview of the event log we performed a dotted chart analysis [2], i.e. plotting the log over time where each event from the log is represented by a dot in the chart. This enables visually examining the log and lets humans quickly identify patterns, that are impossible to see from browsing the log in text format and hard to detect from standard statistics calculated from the log (unless one already knows which pattern he is looking for).

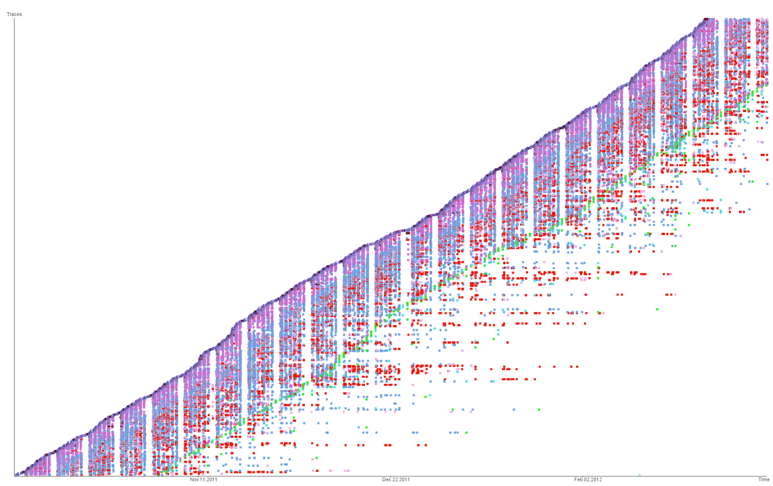

Fig. 1. Dotted chart of the financial log

An overview of the whole log can be seen in Figure 1. The 13.087 process instances (ordinate) are plotted over a period of time from October 2011 – March 2012 (abscissa). The process instances are sorted by the time of their first event, the first process instance starting on the bottom left and the last one in the top right corner (one can see that there is a cut in the end, the last process instance starting on the 29th February). The remaining process instances are cut off in the log with the last event occurring on the 15th March 2012. The chart contains 262.200 dots representing the events. The events are classified so that events belonging to the same process step fall into the same class, each class having a different colour in the chart. As the time period is quite coarse and many events are occurring at similar points in time, there are only five dominant colours to be seen at this resolution and one would have to zoom in order to see all the events.

In the chart there are clusters of six vertical bars to be seen. These bars represent the days of work in the financial organisation, Monday to Saturday. Saturdays are less busy than weekdays and employees seem to be processing more new rather than old applications. The most common activity on Saturdays is calling the customer after an offer has been sent (*Nabellen offertes*, represented by blue dots). Filling in information for the application is also done on Saturdays as well as a few calls regarding missing information, whilst assessing the application is rarely done on Saturdays. The gaps with very few events between the clusters are 1 day long (Sundays), except for Christmas time with a length of two days (Sunday, the 25th, and Monday, the 26th December). The line representing the first events in the process instances (leftmost dot of each trace in the chart) shows no gap on Sundays as customers are able to apply for loans every day (e.g. online), even though the gradient is smaller on weekends. That might result from either a smaller need for loans on weekends or certain channels (e.g. applying in person) being closed to customers on weekends.

The smaller gradient of new applications on weekends can be seen better when zooming in (Figure 2). Here one can also see the two most exceptional

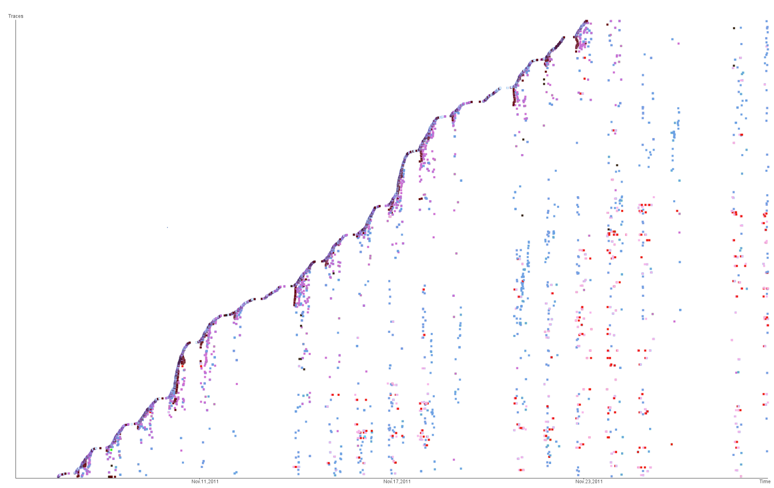

Fig. 2. Zoomed into November

days in the log, 10th and 17th November (Thursdays) which show twice as many applications as normal weekdays. We can't explain that from the log, but stakeholders might be able to identify the stimulating factors for these two days. Mondays are slightly busier as unprocessed applications from the weekend have queued up.

2.1 Activities

A closer look at the line of green dots in Figure 1, shows that it is exactly 31 days away from the corresponding point on the start line of the process instances. Filtering the events of this type one obtains the distribution of the *cancel* activity occurrences (Figure 3).

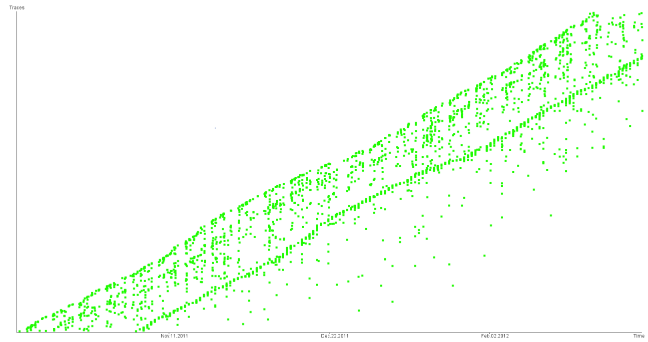

Fig. 3. Cancel activity

By zooming in, we found that most cancellations are made either right after the application, 6-8 days later, or 31 days after the application. 6-8 days later is the time when the customer receives a first call from the organisation to discuss an offer or about incomplete documents in the application. In Figure 1 those calls are represented by the red (*Nabellen incomplete dossiers*) and blue (*Nabellen offertes*) dots. The occurrences of *cancel* 31 days after the application is submitted are likely due to a timeout, as they are always performed by *resource 112*. This resource is performing process steps 24x7 and must therefore be the system that is assigned to such batch activities.

Looking at the activity *declined* (Figure 4), one sees that this activity occurs mostly right after the application has been submitted. In fact 26% of the applications are declined automatically by the system within one minute after an application has been submitted (*variant 1*, containing 3 events). Another 14% of the applications are declined within a few minutes after an employee has had a look at the incoming applications *Afhandelen leads* (*variant 2*, containing 5 events).

So 40% of the cases are declined automatically or with only minimal effort from employees, and contain less than six events. 30% of the cases are longer (up to 15 events), but still belong to a variant that occurs more than once, while another 30% are unique (exceptional behaviour) and consist of more than 16 events.

Figure 5 is also showing the log as a dotted chart with the traces being plotted over time, but this time the traces are sorted by the number of events

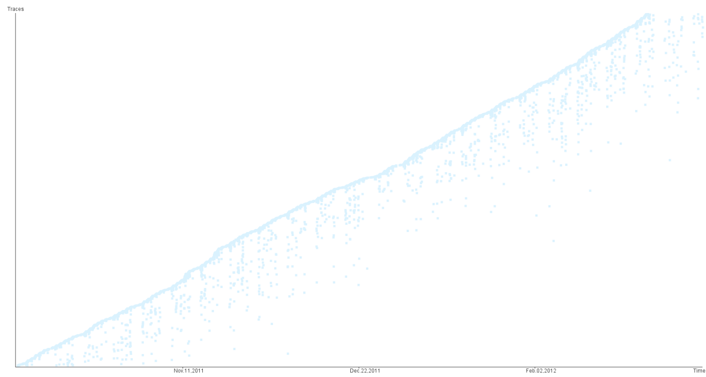

Fig. 4. Declined activity

they contain. Traces with few events show at the bottom of the chart and traces with many events on the top.

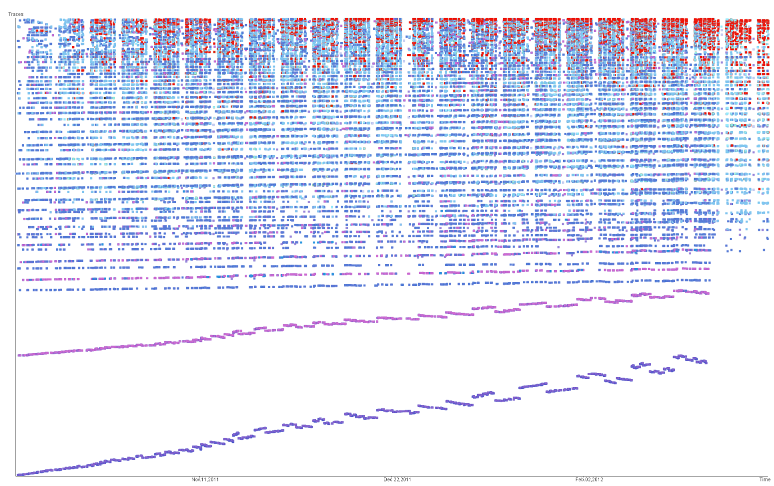

Fig. 5. Traces sorted by number of events

One can see the two above mentioned variants that make up 40% of the log, the blue/purple line on the bottom containing the cases with only 3 events, and the pink line above representing *variant 2*. The bottom right corner of the chart is empty because there are no process instances logged that started after 1st March 2012, therefore there are no short traces showing after this date. The chart also shows that calls about missing documents (*Nabellen incomplete dossiers*, represented by the red dots mostly located in the upper quarter of the chart) are only occurring in the 30% of the cases that are unique and contain more than 16 events. Figure 6 shows the traces sorted by their duration, the shorter traces being plotted at the bottom of the chart.

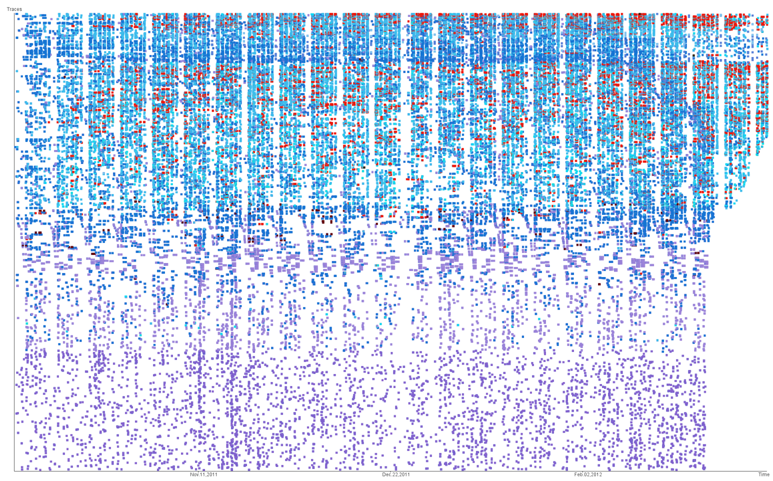

Fig. 6. Traces sorted by their duration

The events coming from the process step *decline* (purple dots) are mostly found in the lower half of the chart, belonging to the process instances that take less than 32 hours. Only 6% of the cases took between 32h and 7 days, 27% between 7 and 23 days. Another 6% fall into the category of applications that have been cancelled automatically after 31 days.

When looking only at the successful applications (there are 2246 cases where the process step *activated* has been performed), only 29 applications have been processed in less than 33h and only 80 in less than 6 days. 98% of the successful applications have been processed in less than 44 days. 75% had a duration of $6-21$ days. In Figure 7 the traces are plotted over the number of events, sorted by the number of events.

The chart shows the distribution of cases according to the length (here measured by the number of events in the case). Cases with many of events show at the bottom while the 26% of the traces that contain only three events (*submitted*, *partlysubmitted*, *declined*) are at the top. Looking at the chart it also becomes clear that *submitted* followed by *partlysubmitted* are the always the first two activities in every case. The lower picture shows the same chart, but this time with only the *approved* activity highlighted. By counting the bars it becomes clear that the approval step is one of the last activities in every process instance and occurs only after at least 17 other activities have been performed before.

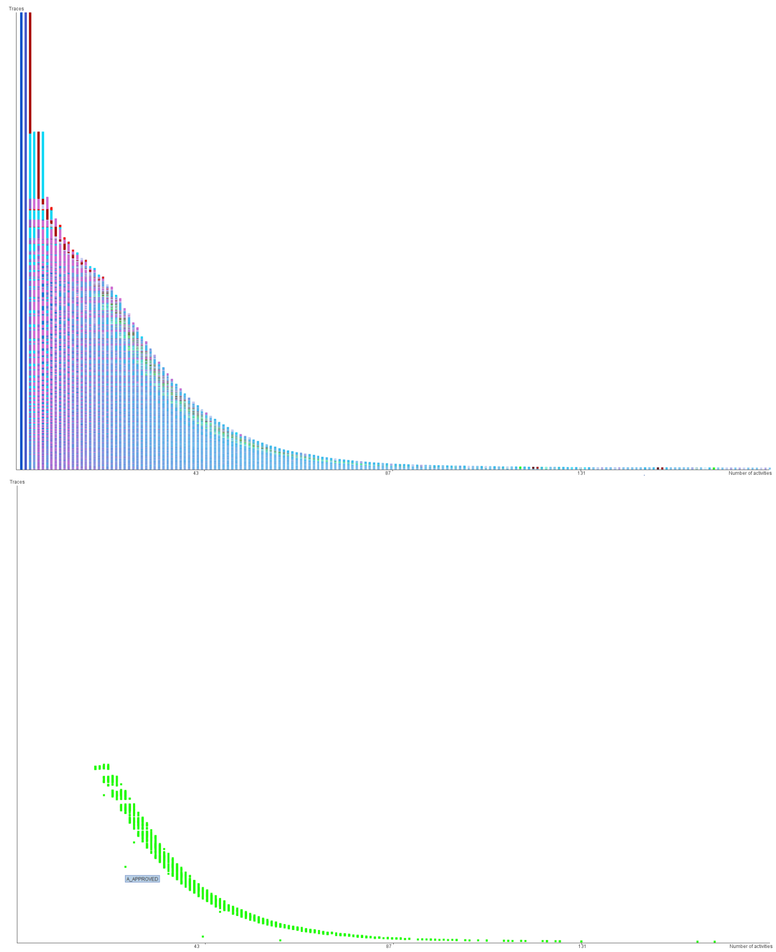

Fig. 7. Traces plotted over their size, sorted by their size

2.2 Resources

Looking at a dotted chart where resources are used for classifying the dots one can see how a resource is involved in the process instances. In Figure 8, the contribution of resource 112 (system) is shown, while the other resources are filtered out.

The system is supporting the application submission, automatic checks (pre-accepting) and automatic cancellations, but it is also scheduling activities that involve human interaction (adding information to the application, fixing incoming leads, and run fraud detection).

Plotting the resources instead of traces in a chart (Figure 9) enables an analysis of how busy certain resources are.

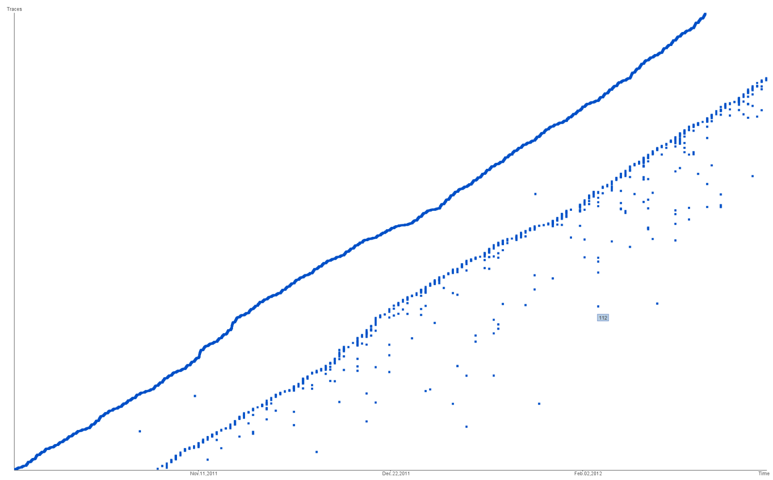

Fig. 8. System resource 112

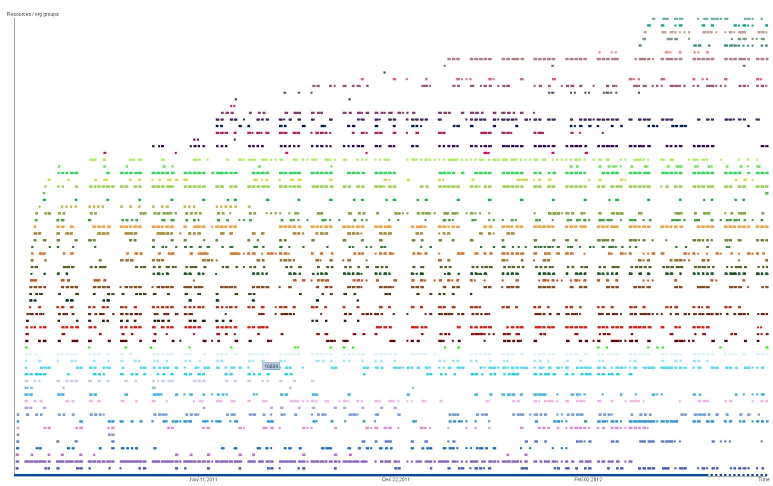

Fig. 9. Resources sorted by first occurrence

In this case the system resource (blue line at the bottom of the chart) is almost constantly processing events while most of the other resources are less busy. For a more accurate analysis one would have to take into account not only the number of events but also the time in-between, especially between start and complete events of the same activity. As the resources are sorted by the time of their first occurrence in the log, the chart shows also which resources started later participating in the process (allows a comparison between old and new employees). In this scenario 20 of the resources did not participate until November or later, 5 new resources joined in February.

Looking at the process steps that the resources were assigned to, we could make the following observations. There are only six resources in charge for approving applications: 10138, 10972, 10629, 10609, 10809 and starting from 7th February also 11289. Exceptions are the resources 11339, 10779 and the system resource 112 which have approved a small amount of cases (respectively 9, 2 and 3 times). Eight resources (10859, 10862, 10880, 10971, 11111, 11120, 11200, 11202) are only showing in process steps where only the complete event is logged, but no information about the duration of the activity is given. Examples of this kind of process steps are *select*, *create*, *sent*, *declined*. All the other resources are *all rounders* and not only performing this kind of tasks but also additional tasks that take a measurable amount of time like phone calls with a customer or assessing an application. Resource 11029 is only performing the *sent back* activity while 10188 performs 78% of the fraud checks.

3 Mining the control flow

The process consists of 3 sub-processes (called W, O and A), each consisting of several activities which carry a label of the process they belong to in their name. Process W describes tasks that are involving human interaction and span across a period of time, e.g. checking for fraud, calling the customer, assessing the application, etc.. Events from the the processes O and A describe results of an action or a decision, e.g. application declined, cancelled, accepted, approved, sent back, finalised, etc.. As a typical process instance includes activities from all three sub-processes, we analysed the whole process and did not split the event log into parts corresponding to the sub-processes. The most common execution of the process is shown in Figure 10. That is the variant that makes up 26% of

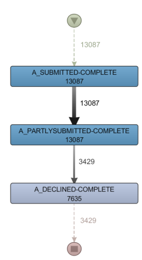

Fig. 10. Most common sequence found in the log

the cases found in the log and describes only the cases in which the application is declined immediately after submission. A model that is not abstracting from the activities involved, but contains a simplified model of the paths in the process is shown in Figure 11. This model includes all the activities occurring in the log and shows three major branches. After submitting an application it can either be declined immediately (middle branch) which corresponds to the above

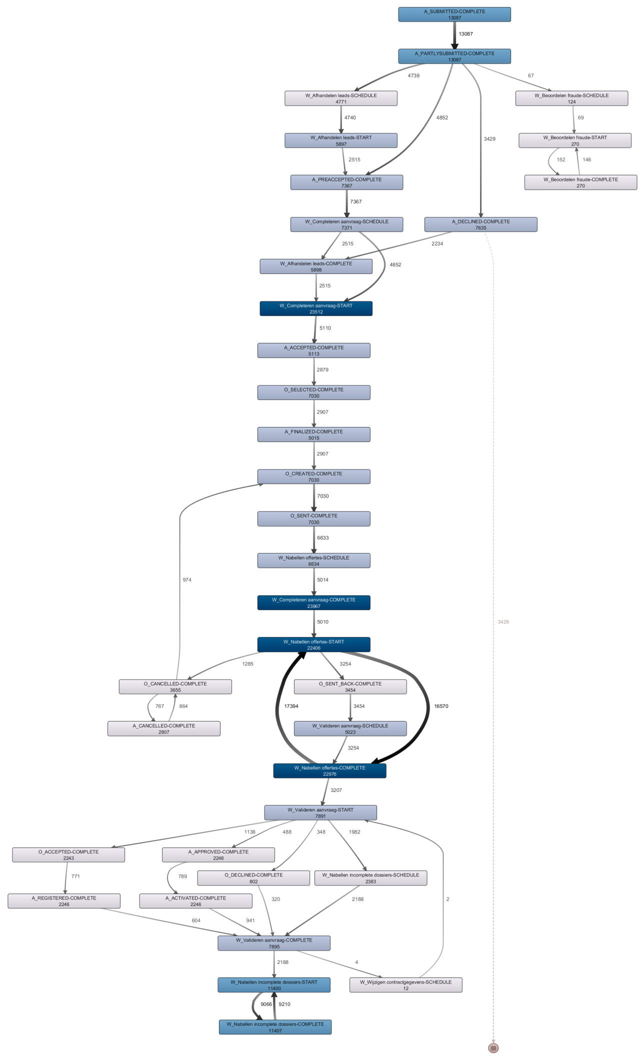

Fig. 11. Fuzzy model of the process with high abstraction level in paths, mined with Disco

mentioned most common sequence, but it can also be examined for fraud (right branch) or follow the path towards acceptance (left branch). In case of the left branch it is either *pre-accepted* by the system or processed by an employee *fixing leads*. After that the application is initially accepted and an employee starts filling in information for it. An offer is created (*O CREATED COMPLETE*) and sent to the customer. After the offer is sent, the customer is called once or more, presumably to discuss the offer. Finally the customer sends the offer back, so it can be assessed. The most frequent and time consuming activity occurring in that phase are phone calls with the customer to clarify issues about missing documents. Based on the assessment the offer can then still be declined or approved (in which case it is always registered and activated). An application is approved 2246 out of 13.087 times in the log. Note that the model in Figure 11 does not allow a path back from phone calls about missing documents. That is only due to the fact that the model abstracts from infrequent paths, which are existing but just not shown.

4 Used Tools

For mining the control flow model, first an implementation of the heuristic miner [4] has been used, which reads the model in XES format [7]. However, after transforming the causal net into our Petri net model [10], the resulting process model was too complex to understand. When trying to analyse the log with ProM 6.1 [9], we ran into performance problems (when using the heuristic and genetic miner plugins as well as the dotted chart plugin) as the log is quite big. Specifically for the challenge Fluxicon provided a project file for the log, which allowed us to analyse the log in the demo version of Fluxicon's tool Disco. Disco provides a fuzzy miner [5] to get an overview of the process without decision nodes or parallel gateways known from other process modelling languages. One can define the level of abstraction in order to abstract from or include infrequent behaviour. It also provides statistics and allows for filtering the log. We used this functionality when analysing the process behaviour for successful and unsuccessful loan applications.

For the dotted charts we used a self-developed prototype which allows for different perspectives, filtering and sorting of the data. We used openXES for reading the log into the chart as well as for calculating some of the statistics like the distribution of the case duration.

5 Conclusion

In this paper we have analysed a log from a loan application process by using dotted chart and control-flow mining techniques. We gained insight not only into structure of the process, but also into the resource structure and the roles within the organisation. While it doesn't make sense to make statements on the overall average processing time of applications, we split the log into parts where it does make sense, the instances where applications are automatically declined and the successful applications. Another possibility which is promising would be to split the log into parts containing clusters of cases with a similar amount of loan requested by the customer and analyse whether the process structure or performance depends on the amount of money.

We found changes in the resource perspective that happen over time and identified days with exceptionally high and low workload.

While the process is easily understood for the most common cases in which the loan application is declined, it shows a high variability for cases where the application is approved. By using the fuzzy miner provided in Disco we were able to abstract from that variability. Nevertheless there is always a lack of insight when looking only at the process log. A better explanation for the results of our analysis and a better understanding of the process could be achieved in discussion with the stakeholders, who have insight into the process and the business environment. While process mining reveals the real process behaviour, the most common paths as well as the exceptions, it would be interesting to know the actual intended behaviour.

References

1. W. M. P. van der Aalst, A. Weijters, and L. Maruster, *WorkflowMining: Discovering ProcessModels from Event Logs*, IEEE Transactions on Knowledge and Data Engineering. 16(9):1128-1142, 2004.
2. W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011.
3. A. Rozinat and W. M. P. van der Aalst, *Conformance Checking of Processes Based on Monitoring Real Behavior*, Information Systems. 33(1): 64-95, 2008.
4. A. J. M. M. Weijters and J. T. S. Ribeiro, *Flexible Heuristics Miner (FHM)*, IEEE Symposium on Computational Intelligence and Data Mining (CIDM). 2011.
5. C. W. Guenther, W. M. P. van der Aalst, *Fuzzy mining: adaptive process simplification based on multi-perspective metrics*, BPM 2007 Proceedings of the 5th international conference on Business process management. 2007
6. M. Song and W.M.P. van der Aalst, *Supporting Process Mining by Showing Events at a Glance*, Seventeenth Annual Workshop on Information Technologies and Systems (WITS'07). pp.139-145 2007
7. C. W. Guenther, *Extensible Event Stream*, http://www.xes-standard.org/_media/xes/xes_standard_proposal.pdf. 2011 (accessed 14/02/2012).
8. B. F. van Dongen, *BPI Challenge 2012 - Event log of a loan application process*, http://data.3tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f. Eindhoven University of Technology, 2012.
9. W. M. P. van der Aalst, B. F. van Dongen, *ProM : The Process Mining Toolkit*, Industrial Engineering. 489: 1-4, 2009
10. U. Winkler, W. Gilani and A. Marshall, *Business Driven BCM SLA Translation for Service Oriented Systems*, 16th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault Tolerance. Springer, 2012