Identifying Users' Skill Level through the Process Mining of Software Logs

Kjell Spijker
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
k.spijker@student.utwente.nl

ABSTRACT

These days, information systems generate extraordinary amounts of data. A very useful metric for improving these information systems is the skill level of a user. Therefore, this research will demonstrate how different skill levels of users can be identified, based on a software generated request log.
After pre-processing the raw data of a request log, process mining will be applied to the resulting event log to generate models, which will be used to identify differences between different skill levels of users. Therefore, this paper describes in a way to identify different skill levels of users and it describes how to extract those levels from software logs. Furthermore, this paper illustrates how these results could be applied in order to improve the currently existing software system.

Keywords

Process mining, request log, event log, user skill, user behaviour

1. INTRODUCTION

Since the rise of personal computers, software has begun to play an increasingly important role in both our private and our professional lives. Within many companies and organizations, software has taken over key roles. Despite the important role, many users might not realize the full extent of what is possible with the software that they are using. This means that the company or organization might improve their processes by ensuring that all users use the software to its' fullest extent.

To be able to improve such processes, knowing the skill level of a user is very valuable. This might, for exmaple, help guide the development of or the training with the software. Therefore, this paper will illustrate a method to identify different skill levels that users of the software might have by applying process mining.

Whilst process mining might not be the only possible solution for this problem, another possibility might be using neural networks, it might be an easier alterative than other possible solutions. This is why this research will use process mining to identify users by the different skill levels they have, based on the request logs generated by Gidso Regie [1].

This paper will first look at what data, taken from logs generated by Gidso Regie, is relevant to use in process mining. It will then look at the differences that exist between different skill levels of users, followed by how those differences could be used to categorize users. Finally, this paper will show how the results of the process mining could be used in order to improve the software that generated the logs.

2. RESEARCH QUESTIONS

This paper will address the following research questions:

RQ 1 How can different skill levels of users be determined by process mining software logs generated by Gidso Regie?

RQ 1.1 How must the data be transformed for it to be useful for process mining?

RQ 1.2 What differences are there between skilled and unskilled users?

RQ 1.3 How should those differences be used to categorize users?

RQ 2 How can we use the difference between skilled and unskilled users to improve the process of the users of Gidso Regie?

3. BACKGROUND

3.1 Gidso

Like with all types of institutions, software plays an increasingly important role within municipalities. In order to help those municipalities tackle the challenges of helping their citizens optimally, Topicus Overheid created the Gidso tool.

Gidso acts as a guide for municipalities which lists where what help can be found, who offers that help, what services are available and what problems are active in certain areas of the municipalities. During this research, the main focus will be on a specific part of Gidso called Gidso Regie, which includes most of the main functionality of this tool when it comes to direct user interaction.

3.2 Process Mining

Process Mining is a series of techniques that "aims to discover, monitor, and improve actual processes by extracting knowledge from event logs" [9] generated by systems. At the basis of process mining are models known as process

31th Twente Student Conference on IT July 5th, 2019, Enschede, The Netherlands.

[1] https://gidso.nl

models. Process models are a graphical representation describing processes, organizations, and products.

These models are extracted from an event log. These event logs are logs which have a few minimum requirements in order to be useful for process mining. Firstly, an event must relate to an "activity" and be part of a certain "case". Secondly, there must be some way to order the events based on when they occur. [1] Having gathered data in such event logs, process mining can then be used in order to construct models without any "a priori" knowledge [9]. The model that process mining generates, depends on which of the different algorithms have been used.

3.3 Data format

As mentioned in the above section, for process mining to work, an event log is required to include timestamps, activities, and cases. Since the Gidso application was not built with process mining in mind, the logs it automatically generates are not perfectly suitable for process mining without some pre-processing. The format that the raw log data takes can be seen in table 1 on page 6.

3.4 Definition of skill

In order to specify what this research considers to fall under skill, a definition of skill is needed. Since this research will try to determine how well different users can use the software system, the following definition is taken from the Oxford Dictionary:

> **Definition 1.** Skill: *The ability to do something well; expertise. [16]*

Throughout this paper, this definition of skill will be assumed.

4. RELATED WORK

This research encompasses two concepts that form their own research areas: process mining and the clustering of users based on their skill level. Whilst there is plenty of research available on either of the given concepts, research that actually combines the two is scarce. These concepts are not very tightly related, but process mining plays more of a supportive role in the clustering of users (i.e. this research will use process mining to cluster the users). The research into one of these concepts will still be very valuable to this work.

4.1 Process Mining

In order to determine if process mining can be used for the purposes that this research intends, this work will use some existing techniques that have proven to be useful in the field of process mining. Most applications of and research into process mining use data in the form of event logs from process aware applications [5]. Since Gidso has not been built with a process-aware mindset, it does not produce logs based on process events, but on events that occur on the server. More specifically, Gidso uses the Apache Wicket Framework[2], which generates the request log. A thesis that bears some resemblance to this research has been written [3]. Although this thesis looks specifically to user loyalty, it uses data partially generated by the same framework and shares similarities with identifying "usage patterns which might be of influence" [3].

More general research into the field of process mining has also been done [1]. Process mining has also been applied in a set of different fields, including auditing [11], insurance [19], and healthcare [14] [17]. Even the process mining of software logs has been done before [18], although it is not as prevalent as some of the other areas. Other methods to analyse software systems more often target the runtime perspective instead of the user [2] [8] [13].

A lot of work has also been done in the field of web usage mining [15] [4]. Where existing approaches use statistical and/or data mining methods to analyse user behaviour in web environments, process mining can be seen as an important improvement in the field of web analytics [18].

4.2 User Clustering

Another part of this research is how the users can be clustered based the level of skill they have with the software. The closest other research comes to combining this with process mining, is the thesis mentioned above [3]. Apart from this, research has been done, albeit without process mining, into the clustering of user behaviour based on web log data [12]. Other research that analyses users' activity has also been done [20].

The behaviour of users is also relevant to their clustering. Research into this has been done on several occasions [10] [6]. In those cases, the research was focused on extracting behavioural user data from logs generated by a web server, similar to how Gidso generates logs. Although these papers share some similarities with this research, they do not use process mining and do not try to identify skill levels of users.

Research specifically into the detection of users' skill levels has been done in [7]. In this research, a number of attributes have been identified that helped machine learning algorithms determine the skill level of a user. Similar tactics for the choosing of such attributes might be applicable to this work.

5. METHOD

The methods used in this research consist of several parts. First, the raw log data needed to be processed in such a way that it could be used with process mining tools. This happened during the pre-processing of the dataset. Once the data was in a usable format, the process mining tools were used in order to generate models that could be studied during the next steps. This model was then analyzed and the key points for discovering user skill levels were determined. Finally, the process enhancement stage used those results in order to find points on which to improve the currently existing process.

5.1 Pre-processing

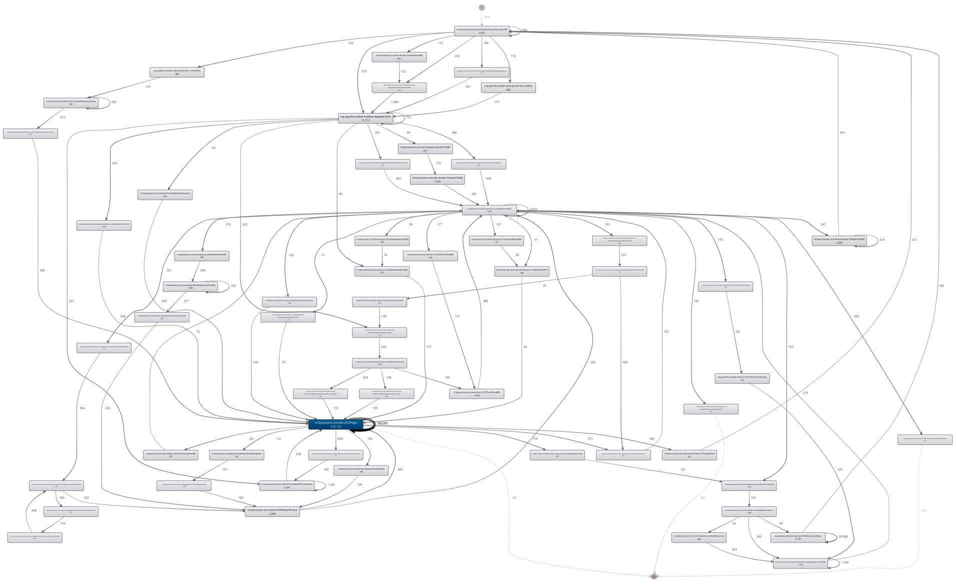
The data generated by the Apache Wicket framework is, like most software generated logs not made for that purpose, not directly suitable for process mining. Therefore, all relevant data will first be extracted and prepared in a format that the various process mining tools can understand. The raw format that the log follows is explained in table 1 on page 6.

Besides parsing the data to a format that can be used in process mining, the pre-processing phase also takes care of anonymizing the data. Whilst parsing, random strings are created on a per user basis, in such a way that each user has exactly one random string associated with that user. These strings will replace their respective usernames and, during process mining, they will be used as a case identifier. This way, the structure of the data will not change, but the users' right to privacy gets respected.

After extracting the timestamp and the anonymized case identifier, the action that the user took will be extracted

[2] `https://wicket.apache.org`

Figure 1. An example of the discovered model. Generated with 18.8% of activities and 1% of paths



from the JSON object generated by wicket. This action is retrieved from the component class in the request object. This class represents the component on the page that the user interacted with, this can, for instance, be a button to create a new dossier within Gidso Regie.

Once the three pieces of information that process mining tools need are known, these are stored in a CSV file that will be used in the following phases. An overview of the data format used in the CSV file can be seen in table 2.

5.2 Process Discovery

With the data ready for process mining, it will be loaded into the tools. As far as tools go, there are several options that were attempted during this research, namely Disco [3] and ProM [4]. Whilst ProM contains many different extensions and algorithms to use, Disco gave, at least for this dataset, models that were clearer and more useful when analyzing these models.

The dataset as generated by Apache Wicket contains many different paths that can be followed, so this paper will focus only on the parts that are relevant for determining skill levels. Also, given the confidentiality of the dataset, similar data has been created in order to replicate the structure of the models of the original data, but also to allow for publication. An example of a complete model can be seen in figure 1.

5.3 Process Analysis

Once the complete model has been discovered, possible points need to be found where skill levels of the users can be extracted from. The points on which this occurs most clearly, are the points where there are multiple paths from the same origin activity to the same destination activity. These different paths can, in some cases, identify differences between skill levels of the users taking those paths.

Besides directly analyzing the model, a case by case view might be required in order to identify which users belong to which group of skill level. Disco, for example, generates different variants based on the order of events that a user has taken and groups users into these variants. This can then be used to identify which users belong to which group. More complex logs, however, contain a more unique sequence of events possibly rendering this functionality useless when all variants only contain a single case. An example of this can be seen in figure 4.

6. RESULTS

6.1 RQ 1.1: Data transformation

Process mining always requires a few key items to be included in the dataset. First, there must be a case identifier, which, in the case of Gidso Regie, is the anonymized string representing a single user based on the username in the raw request log. Second, process mining requires a timestamp in order to determine in which order the recorded events occurred. These are already included in the request log and can, therefore, easily be reused for this purpose. Last, an activity is required in order to know what the current state of the process is. These are extracted from the JSON object that Apache Wicket generates, which contains a request object with an eventTargetLog object and finally the componentClass key which, as value, holds the component of the page that the user interacts with, e.g. clicking a certain button or link.

All this information can then be exported to a CSV file, which can be used as input for either Disco or ProM. An example of a part of the model that Disco generated based on the dataset can be seen in figure 1.

6.2 RQ 1.2: Finding skill level differences

Now that the model has been made, it can be used to find points where there is a difference between the skill levels that users have. Keep in mind that whilst there are patterns to look for, these patterns are not guaranteed to indicate differences between skill levels of users.

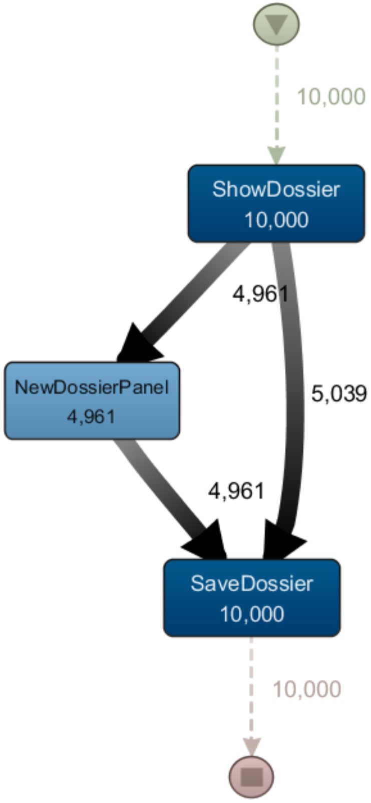
The patterns that have been found to be likely to identify differences between skill levels are small subsections of the original model produced by the process mining tool during Process Discovery. This subsection contains a single action, spread over multiple paths, i.e. paths starting in the same activity and ending in the same activity, but go from start to end via a different set of activities. These different paths might identify different skill levels, however, which

[3] https://fluxicon.com/disco/

[4] http://promtools.org/

path signifies more skill is still up to the person analyzing the model. An example of such paths can be seen in figure 2.

Figure 2. An example of a subsection of the discovered model



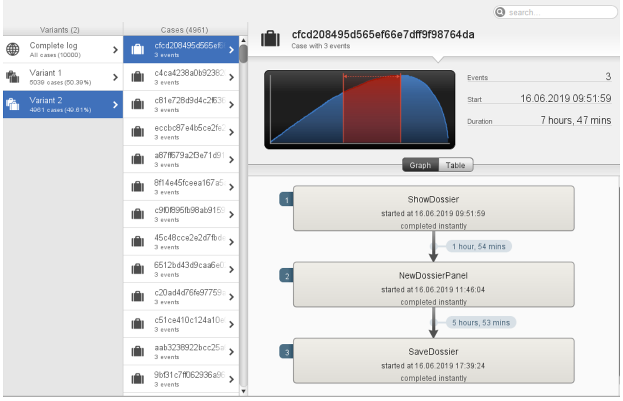
In the example in figure 2, there are two paths visible. The first activity, which both paths share, shows the details of a dossier to the user. From here, users can take one of the two paths in order to make changes to that dossier. Users that have less skill in using the software open the ”NewDossierPanel” and make their changes there, whereas more skilled users generally edit the data directly in the ”ShowDossier” screen. After either of those actions have been completed, the users save the updated dossier, which results in the ”SaveDossier” activity, which is again shared between the paths. This example illustrates clearly that a difference can be observed between more skilled and less skilled users in these sort of subsections of the process model, but that human intervention is still required in order to make the decision of which path is more skilled than the other.

When comparing the example in figure 2 with the complete model in figure 1, it becomes clear that it depends on the dataset how usable the models are. Considering that figure 1 only shows 18.8% of all activities and 1.0% of all possible paths, it shows that models generated from data not made for process mining, might become immensely complex. As such, finding the different paths as described above becomes increasingly more difficult when the model contains more variations.

6.3 RQ 1.3: Categorizing users

Using the differences found under research question 1.2, users can now, after determining which path the more skilled users take, be clustered into their respective categories. These users can be categorized using either the ProM or Disco process mining tools, where Disco again is easier to work with. Using Disco, looking at the cases and identifying which of the variants contains the path with the required skill level brings up a list of all cases with that skill level. In a larger model than the example shown in figure 2, the list of variants grows very quickly and this process then becomes extremely time consuming for the person identifying the skill levels of users. For example, doing this for the original dataset gives a list of 665 variants, all containing a single case. Therefore, this method for categorizing users is not much better than doing this entirely by hand, based on the data generated in research question 1.1.

Figure 3. The cases view in disco



search...

Variants (2)
Complete log
All cases (10000)
Variant 1
5039 cases (50.39%)
Variant 2
4961 cases (49.61%)

Cases (4961)
cfcd208495d565ef6
3 events
c4ca4238a0b92382
3 events
c81e728d9d4c2f636
3 events
eccbc87e4b5ce2fe2
3 events
a87ff679a2f3e71d91
3 events
8f14e45fceea167a5
3 events
c9f0f895fb98ab9159
3 events
45c48cce2e2d7fbde
3 events
6512bd43d9caa6e0
3 events
c20ad4d76fe97759a
3 events
c51ce410c124a10e
3 events
aab3238922bcc25a
3 events
9bf31c7ff062936a96
3 events

cfcd208495d565ef66e7dff9f98764da
Case with 3 events

Events 3
Start 16.06.2019 09:51:59
Duration 7 hours, 47 mins

Graph Table

1 ShowDossier
started at 16.06.2019 09:51:59
completed instantly

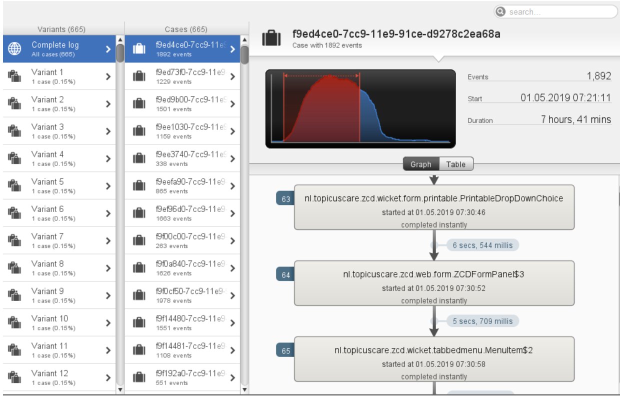
1 hour, 54 mins

2 NewDossierPanel
started at 16.06.2019 11:46:04
completed instantly

5 hours, 53 mins

3 SaveDossier
started at 16.06.2019 17:39:24
completed instantly

Figure 4. The problem with many unique combinations of activities



search...

Variants (665)
Complete log
All cases (665)
Variant 1
1 case (0.15%)
Variant 2
1 case (0.15%)
Variant 3
1 case (0.15%)
Variant 4
1 case (0.15%)
Variant 5
1 case (0.15%)
Variant 6
1 case (0.15%)
Variant 7
1 case (0.15%)
Variant 8
1 case (0.15%)
Variant 9
1 case (0.15%)
Variant 10
1 case (0.15%)
Variant 11
1 case (0.15%)
Variant 12
1 case (0.15%)

Cases (665)
f9ed4ce0-7cc9-11e9
1892 events
f9ed73f0-7cc9-11e9-
1229 events
f9ed9b00-7cc9-11e9
1501 events
f9ee1030-7cc9-11e9
1159 events
f9ee3740-7cc9-11e9
338 events
f9eefa90-7cc9-11e9-
865 events
f9ef96d0-7cc9-11e9-
1663 events
f9f00c00-7cc9-11e9-
263 events
f9f0a840-7cc9-11e9-
1626 events
f9f0cf50-7cc9-11e9-
1978 events
f9f14480-7cc9-11e9-
1551 events
f9f14481-7cc9-11e9-
1108 events
f9f192a0-7cc9-11e9-
551 events

f9ed4ce0-7cc9-11e9-91ce-d9278c2ea68a
Case with 1892 events

Events 1,892
Start 01.05.2019 07:21:11
Duration 7 hours, 41 mins

Graph Table

63 nl.topicuscare.zcd.wicket.form.printable.PrintableDropDownChoice
started at 01.05.2019 07:30:46
completed instantly

6 secs, 544 millis

64 nl.topicuscare.zcd.web.form.ZCDFormPanel$3
started at 01.05.2019 07:30:52
completed instantly

5 secs, 709 millis

65 nl.topicuscare.zcd.wicket.tabbedmenu.MenuItem$2
started at 01.05.2019 07:30:58
completed instantly

As shown in figure 3, disco can give a clear overview of the different variants (based on the example subset data), which can be considered different skill levels. As mentioned in the previous paragraph, however, this can quickly grow out of control with complete request logs. This is demonstrated in figure 4 and clearly shows that the variants all contain a single case with up to 2000 events, making the categorization quite difficult.

6.4 RQ 2: Software enhancement

The results of research question 1 and its sub-questions can be used to try to improve the software from which it has been generated. The first way this can be done, is by using process mining quite directly. The models have been generated in the previous steps and can, therefore, be used to attempt an improvement to the existing model. For example, some paths or variations of paths between two events that the model contains might not be the most optimal way to use the software and could, therefore, be considered to be changed or removed altogether. Furthermore, developing certain parts of the software might be given priority if more skilled users run into problems as opposed to when less skilled users run into problems, in which case users might more easily make a mistake instead of the system being broken.

Another way to offer improvements is not by changing the software, but by improving the accompanying training modules. Especially for users with a lower skill level, these training modules might improve the way that they interact with the system and thus improve their skill level. Because of the results of research question 1, the training modules can now be offered much more precisely to only the users of the software that need those parts of the training.

6.5 Discussion

The advantages of the methods used in this paper include the fact that gathering data about the skill levels of users is very valuable to both the development of the software itself, as well as the development of the accompanying training sessions. The process that a user, or a majority of users, takes might also influence further development of the software. This allows developers to better choose how their product will take shape in the future.

The models that are used in order to gain these advantages, however, have the potential of being both extremely useful or hopelessly confusing for people. Models generated from a good dataset will be more useful than confusing, but this might not always be achievable in the real world. Especially working with (request) logs that have not explicitly been made for process mining, the resulting models contain so many different sequences of paths that the models become either incomplete or more confusing than useful. Trying to extract the required key events on which skill levels can be determined is, after all, still a human task within process mining and becomes more complex as the model becomes more complex.

7. CONCLUSION

In this work, a method has been described that can be used to extract the data required for process mining out of logs generated by software, in this case a request log from Apache Wicket. It has shown that this data could then be used to discover models using process mining tools. Using these methods, users can be categorized into different skill levels based on the different paths existing within the model. Finally, this work has shown were possible improvements might be for the software that these models have been created from.

This research has also shown that while it is possible to identify skill levels of users from software logs using only process mining, it might not be the most efficient method of solving this problem. A next step could focus on using the paths that can be identified using this method in order to create algorithms that categorize users based on those identified paths, possibly even without any human interaction being required.

8. REFERENCES

[1] W. v. d. Aalst. *Process Mining.* Springer Berlin Heidelberg, 2016.

[2] G. Ammons, R. Bodík, and J. R. Larus. Mining specifications. *ACM SIGPLAN Notices*, 37(1):4–16, Jan. 2002.

[3] K. M. E. Assal. Predicting user loyalty in an education support web application based on usagedata. Master's thesis, University of Twente, Apr. 2018.

[4] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlos. Are web users really markovian? In *Proceedings of the 21st international conference on World Wide Web - WWW '12*, pages 609–618. ACM Press, 2012.

[5] B. F. v. Dongen. A meta model for process mining data. In *Proceedings of the CAiSE WORKSHOPS*, 2005.

[6] M. H. A. Elhiber and A. Abraham. Access patterns in web log data: A review. *Journal of Network and Innovative Computing*, 1:348–355, 2013.

[7] A. Ghazarian and S. M. Noorhosseini. Automatic detection of users' skill levels using high-frequency user interface events. *User Modeling and User-Adapted Interaction*, 20(2):109–146, June 2010.

[8] R. Gombotz, K. Baïna, and S. Dustdar. Towards web services interaction mining architecture for e-commerce applications analysis. 2005.

[9] M. C. Hol. Using user workflow analysis in content-intensive applications: Combining process mining and model-driven engineering to create a reusable, scalable and user-friendly solution. Master's thesis, University of Twente, Nov. 2018.

[10] M. Jafari, F. SoleymaniSabzchi, and S. Jamali. Extracting users'navigational behavior from web log data: a survey. *Journal of Computer Sciences and Applications*, 1(3):39–45, May 2013.

[11] M. Jans, M. Alles, and M. Vasarhelyi. The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems*, 14(1):1–20, Mar. 2013.

[12] S. P. Karthik and A. Sheshasaayee. Clustering of user behaviour based on web log data using improved k-means clustering algorithm. *IJET*, 8:305–310, 2016.

[13] D. Lorenzoli, L. Mariani, and M. Pezzè. Automatic generation of software behavioral models. In *Proceedings of the 13th international conference on Software engineering - ICSE '08*, pages 501–510. ACM Press, 2008.

[14] R. Mans, M. Schonenberg, M. Song, W. v. d. Aalst, and P. Bakker. Process mining in healthcare - a case study:. In *Proceedings of the First International Conference on Health Informatics*, pages 118–125. SciTePress - Science and and Technology Publications, 2008.

[15] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, Aug. 2000.

[16] Oxford Dictionary. Skill. *Retrieved at:* `https://www.lexico.com/en/definition/skill`.

[17] A. Rebuge and D. R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2):99–116, Apr. 2012.

[18] V. A. Rubin, A. A. Mitsyuk, I. A. Lomazova, and W. M. P. v. d. Aalst. Process mining can be applied to software too! In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '14*, pages 1–8. ACM Press, 2014.

[19] S. Suriadi, M. T. Wynn, C. Ouyang, A. H. M. ter Hofstede, and N. J. van Dijk. *Understanding Process Behaviours in a Large Insurance Company in Australia: A Case Study*, volume 7908, pages

449–464. Springer Berlin Heidelberg, 2013.

[20] F. Xhafa, S. Caballe, L. Barolli, A. Molina, and R. Miho. Using bi-clustering algorithm for analyzing online users activity in a virtual campus. In *2010 International Conference on Intelligent Networking and Collaborative Systems*, pages 214–221. IEEE, Nov. 2010.

Table 1. Data format of the raw request log

Name	Data type	Description
Timestamp	DateTime	Time at which the request was received by the server
Thread ids	String	Several strings that identify threads and applications that handled the request
User	String	Username of the person making the request
JSON formatted request information	JSON	JSON object of information that wicket generated. The JSON contains, among others, the URL of the request, the time it took for the request to be handled, the button on the page that was used to initiate the requests, etc

Table 2. Data format of the data after the preprocessing phase

Name	Data type	Description
Timestamp	DateTime	Time at which the request was received by the server
Case Identifier	String	Anonymized random string, based on the username
Activity	String	Name of the activity that the user did