Problems and Challenges When Implementing a Best Practice Approach for Process Mining in a Tourist Information System*

Marian Lux and Stefanie Rinderle-Ma

Faculty of Computer Science, University of Vienna
{marian.lux,stefanie.rinderle-ma}@univie.ac.at

Abstract. The application of process mining techniques for analyzing customer journeys seems promising for different stakeholders in the tourism domain, i.e., the tourism providers are enabled to, e.g., find nice offers or partner services and the guests can improve their holiday experience. One precondition for mining processes (high quality) logs. This paper reports on experiences in implementing a data warehouse component for storing process logs in the tourism information system *oHA*. It shows which analysis questions can be answered by applying process mining and analysis on the logs. Finally, lessons learned are discussed.

Keywords: process mining, customer journey, data warehouse, tourism-information system

1 Introduction

This business case shows how we designed a sustainable and scaleable data warehouse architecture as well a log concept for the tourism-information system "*oHA*" (online Holiday Assistant)[1], which provides information and digital services for tourists. In more detail, *oHA* is a digital e-service, accessible for the tourist in form of a web app, mostly in a public WiFi, which is designed for tourism agencies and hotels to make more revenue with guests and provide a better service level to their guests. Fig. 1 shows three examples how the web app *oHA* looks like for a tourist guest. In the following we explain some technical details about *oHA* and go through the three displayed screenshots. This is important for understanding our application terminology and thus for understanding our log concept, later in this work.

The first screenshot shows the main menu of *oHA* with possible menu items to be selected by a guest. Every menu item corresponds to at least one digital service in *oHA*. There are lots of services in *oHA* and to name some of them, a

[1] https://www.luxactive.com/

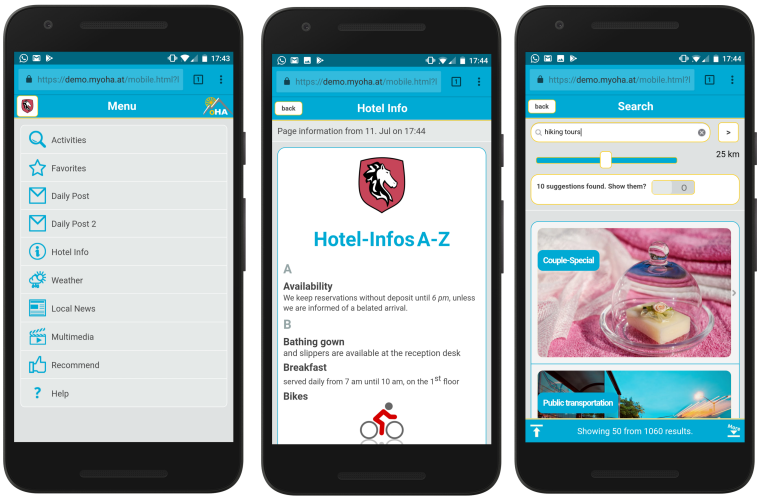


Fig. 1: *oHA* web app for tourist (using MockUPhone `https://mockuphone.com`).

service can be a hotel information (second screenshot), a activity search (third screenshot), a daily post, a regional news, the weather, or a GPS navigation. For instance on the first screenshot, by selecting *Hotel Info*, the second screenshot and by selecting *Activities*, the third screenshot shows up. Each of the displayed screenshots shows a different view in the application which has technically a place name for the current displayed view and we name the statistics behind that, *place usage*. The first screenshot shows, e.g., the place name *HomePlace* and the third *SerachActivityPlace*. On the third screenshot, a semantic search function for touristic activities is provided. The tourist can search for location based and time related *activities* like events near by, POIs (points of interest) or tours to navigate with *oHA*. We record the user entered *search terms* and try to generate processes out of the users search behavior (*search process*) with our stored data which will be covered in more detail later.

Analyzing the guest behavior is an opportunity to distinct *oHA* from competitive tourism information systems. For this reason the *CustPro*[2] project was initiated between the company LuxActive and the University of Vienna. Some of our presented concepts and techniques are already blueprinted and developed and others are still in development. With this work, we will show how we solved the main challenges when starting to implement process mining in a tourist information system. As first step, we designed a data warehouse for storing and preparing logs, to discover further research fields like process mining, aiming to analyze the customer journey process through the tourism platform *oHA*. Fig. 2 shows the different stages of a customer journey in the tourism domain and how it could be interoperated with process mining.

[2] `http://cs.univie.ac.at/project/custpro`

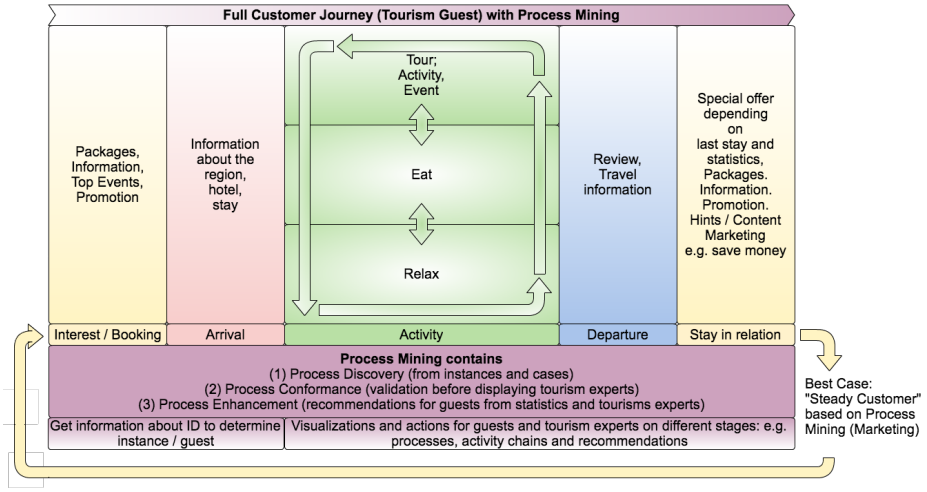


Fig. 2: Implementing process mining along the customer journey with different stages.

The first stage *Interest / Booking* bears the challenge to figure out preferences of the guests for booking a stay. The next stage *Arrival* is for providing all relevant information for the stay which is relevant for the individual guest. In stage *Activity* it is important to provide suggestions for individual activities and an easy way for consuming and booking them. In the stages *Departure* and *Stay in relation* it is important to get feedback about the stay and to encourage the guest and his surrounding people for booking again. For the latter, individual content marketing can can be one method for achieving recurring bookings. Today *oHA* focuses strongly on the stages *Activity* and *Departure* but in future we want to cover all stages of the customer journey with *oHA*. As described before, every stage has different characteristics which require research and implementation. Also recorded logs from the different stages may influence each other. For example, recorded logs from the stage *Activity* may have influence to the stage *Interest / Booking* by serving the right information for promotion, out of historical data from guests.

The first step is to answer the following business process related questions based on the stages *Activity* and *Departure* for customers of *oHA* as the information can be useful for tourism companies when searching for niches, business partners, or increasing their revenue by providing new activities for tourists.

- Which digital services are used by guests mostly?
- Which searched activities like tours, events or POIs are most interesting for guests at the stay and after stay?
- What is a typical search process of a guest (cf. Fig. 3) and how to display it?
- When are the guests searching for services and activities and what are their peak periods?

• Which services and activities are missed by the guests?
• Which services and activities are mostly liked by the guests?
• Who will be a best fitting strategic partner for providing services and activities, e.g., a tour guide?

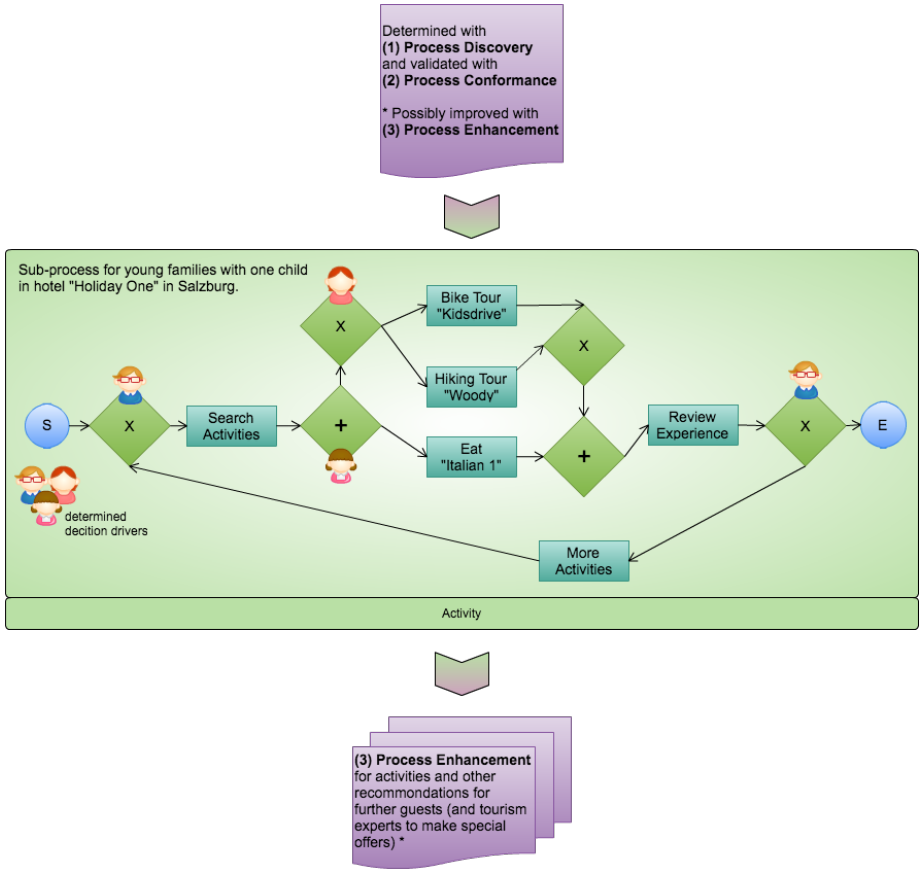


Fig. 3: A sample case on the stage *activity*.

The problem is that the original log implementation in *oHA* had no cases to answer questions on different levels and views. Hence it was not possible to mine processes from the level of individuals, because we could not distinguish between different guest devices. Furthermore, all logs were distributed in files on different local file systems. Thus, log preparations and modification tasks were time consuming and the logs were hard to access due to security restrictions on different servers. Also state changes in our system, which might influence the user behavior, were not recorded and thus taken into account by the logs. Such changes could be for instance hidden or shown menu items in the main menu or

new data sources for *oHA*. Lastly, we had no high-quality maturity level of our logs, which is recommended for instance by the process mining manifesto [1], before starting process mining with logs.

2 Related Work

A literature review suggested process mining [2, 1] as promising technology for answering user behavior related questions as described in the introduction. The preconditions for applying process mining techniques are (high quality) process logs which are challenging to provide in existing systems [3]. Observing data from multiple perspectives has been suggested by work on multidimensional structures in process mining (cf. e.g., [8]). Different approaches and best practices on how to design a data warehouse and how to implement ETL phases, also in the context of process logs, exist, e.g., [4], [7], [5], [12]. How important data warehouses are, is also shown in surveys. [10]. Further research and implementations on process mining in the data warehouse, would be to simplify discovered process models [9] and to improve the quality of process logs [3]. Regarding to our employed relational database, further security [11] or process mining approaches [6] can be researched, evaluated and implemented.

3 Methods and Techniques

This section presents design decisions and methods used for enabling process mining and analysis in *oHA*.

The previous situation in *oHA* was, that all logs including the user behavior were distributed over different file systems and in different file formats. So there was no possibility for tracking the user behavior of the tourists in an efficient way, without time-consuming manual interventions in logs on different file systems. Such interventions include manually gathering the logs, modifying them by removing outliners or test data, and converting them into a format which can be used for statistical calculations or process mining. This was overcome by implementing a central hub for our logs, which acts as data warehouse in our application landscape. An overview of our data warehouse architecture with its main workflow is depicted in Fig. 4.

We opted for an extra physical server environment with its own web server for managing and handling the logs in the data warehouse due to several reasons. Processing logs can be resource consuming and our production systems should not be impacted with performance issues because of resources like memory running out. A data warehouse database is designed to answer complex queries rather than performing a high throughput for updating transactions [4]. To use a web server in front of the database brings also advantages in security, because all the data traffic is encrypted and only authorized applications are able to log data and consume them via a defined API. Using a relational database in general for storing the data makes modifying logs easy to perform and data preparation tasks can be carried out with only a few steps by using, e.g., SQL queries. More

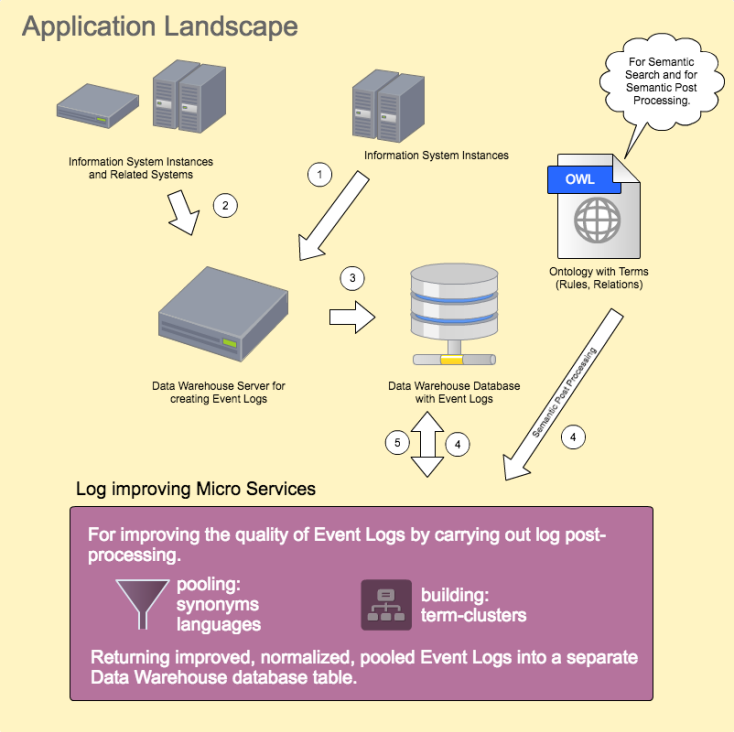


Fig. 4: Overview of the application landscape of *oHA* and the data warehouse. All server-applications *(1)*,*(2)* send their log data to the data warehouse web server. The web server stores the data in a relational database *(3)*. Log improvements are carried out by micro services or other web servers on the data warehouse, which use semantic technologies *(4)* and store the enriched log data back into the data warehouse *(5)*.

technically, we use a Java web server, which is responsible for storing and processing the logs into the database. Our ETL (Extract, Transform, Load) process to the data warehouse is kept simple, because we have the full control over all systems which are logging. So we can also modify our systems which are logging, to fit the needs of the data warehouse. In future, we plan to include also external data from an open world environment like a weather API, tourism databases or a web crawler which is gathering important events nearby. The most notable approach in our current case is the following: If a web client logs events from a tourist, it sends the logs from the client to its responsible server. After that, the server sends the logs to the data warehouse. Due to the mentioned security reasons before, we try to keep our system secure, but with modest effort. For that, we disallow to send the log data directly from a client to the web server of the data warehouse. Only our servers are allowed send log data in JSON format to the data warehouse via our developed REST API with HTTPS.

Regarding the presence of (high quality) logs, in *oHA* some important data was missing, i.e., different cases from a session level to a region, the users language, additional timestamps, and search results of activities from tourists. Thus as shown in Fig. 4 for data transformation and data enrichment processes, the raw log data is processed by separate web servers or micro services. This is also an advantage for a loosely coupled architecture as implemented with our data warehouse web server, which is responsible for the whole data management and communicates via API calls. New web servers or micro services are easy to integrate now into the data warehouse. For instance we currently implement a web service, which uses semantic technologies for handling synonyms and different languages of logged search terms and converts them into a normalized form for improving the quality to further carried out process mining. The processed data is stored back in an extra database table in the data warehouse.

The *oHA* data warehouse database consists of the tables shown in Fig. 5. Because we use an iterative development approach which is still ongoing, not all following presented details are implemented now.

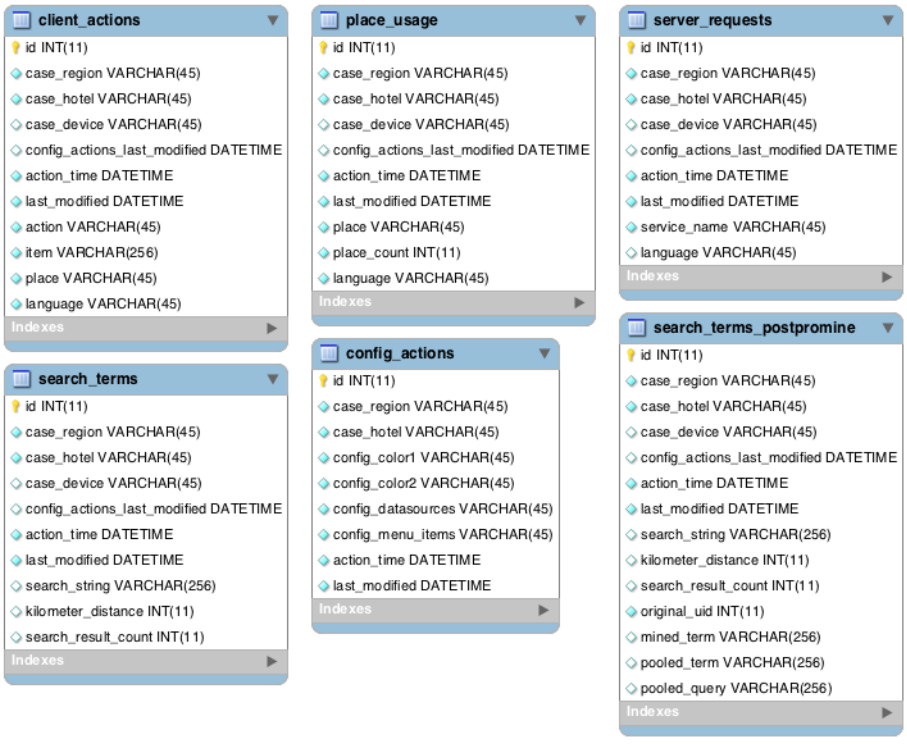


Fig. 5: Relational database tables from data warehouse.

Every table which contains logs has stored cases for region (*case_region*), a customer (*case_hotel*) and a device (*case_device*). For the latter, we implemented a client based solution to store a unique id in the local storage of the clients device which is mostly owned by a tourism guest. This id represents the case for the device and can be also used to identify a session. A session can be calculated together with timestamps of executed actions from the client. E.g, if there is no action with the same device for more than 10 minutes, we can infer a session. Identified sessions can be very useful for process mining. [2] Also current configuration states of the system are recorded with a timestamp field (*config_actions_last_modified*) in every relevant log table. Every time, a system state changes, the timestamp and the system new state will be recorded in the table *config_actions*. A state change is a system configuration change, which impacts the user behavior and thus the recorded logs. The most relevant state changes are currently the change of the displayed menu items, sources for searchable activities or color schemes of the web app *oHA*. So, if e.g., the source for *hiking tours* will be disabled by a tourism provider in the CMS (Content-Management-System) of *oHA*, which is called "*oHA Base*", no tourist can see results after searching for activities which are related with *hiking tours* any more. By executing a SQL query together on both tables, i.e., the table which contains the system states (*config_actions*) and a table of interest for the logs (e.g. *serach_terms*) and by comparing the before mentioned timestamps for a given period, we can identify, in which state the system was, when the logs with the table of interest were created. Thus, this concept enables further improvements in regards to quality and meaningfulness of our logs.

Every client related log table also contains a language field which seems important for performing further analysis tasks. The data warehouse also includes a universal log table *client_actions*, which should log every action from the client in future implementations. It contains three relevant elements. The first is an action type *action*, which could be a selected button or focused text field. The second is the content of the action *item*, which contains, e.g., the title of a selected activity or an entered text. Finally, the third element contains a unique view name, like the *place usage*, form the client *place* for identifying where the action has taken place.

4 Results

With the enriched log concept and the central data storage, most popular search activities and services by tourists in *oHA*, on different locations and in different regions can now be determined. Moreover, analysis questions can be answered from different views due to the different implemented cases, i.e., regions, tourism companies (where every company has its own *oHA* instance), guest devices, and guest sessions. Fig. 6 shows, how mined processes for used digital services in *oHA* on different cases can look like. The first process shows the user behavior of a single user session, the second process shows the same user along the period of

one month and the third process shows, how all users in a hotel used the system in an one month period.

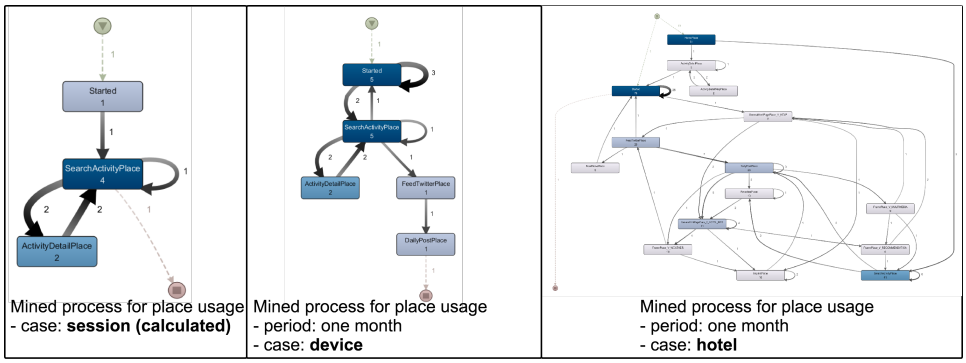


Fig. 6: Different cases for used services in *oHA* (using DISCO `https://fluxicon.com/disco/`).

We can also show which digital services are interesting for the guests on different cases (cf. Fig. 7). The first chart shows statistics from one device, the second shows the same type of statistics from the case of a hotel and the third one from the case of a region. In the first pie chart, the user was most interested in searching for activities. Looking for hotel news was less important. Indirect assumptions about missing services can be derived as well.

In the following, Fig. 8 (left) shows an example for a mined search process of a device which identifies a single guests behavior. One path of the process shows, that the user first searched for a tour and then for different variation of sights. We can also identify peak hours of a day, where guests are demanding different services in our system. This can be a useful information for coordination tasks in service for tourism companies. Fig. 8 (right) shows such an example how peak hours can look like on a hotel, after investigating logs for one month in the data warehouse. At noon, there was most demand of the service *oHA* and thus guests were looking for information.

5 Lessons Learned and Future Work

This work reported on the implementation of a data warehouse in a tourist information system. The primary goal was to improve the quality of logs for analysis tasks such as process mining and to finally understand the customer journey for tourism companies. This can help them in developing attractions, marketing activities, and finally finding their niches.

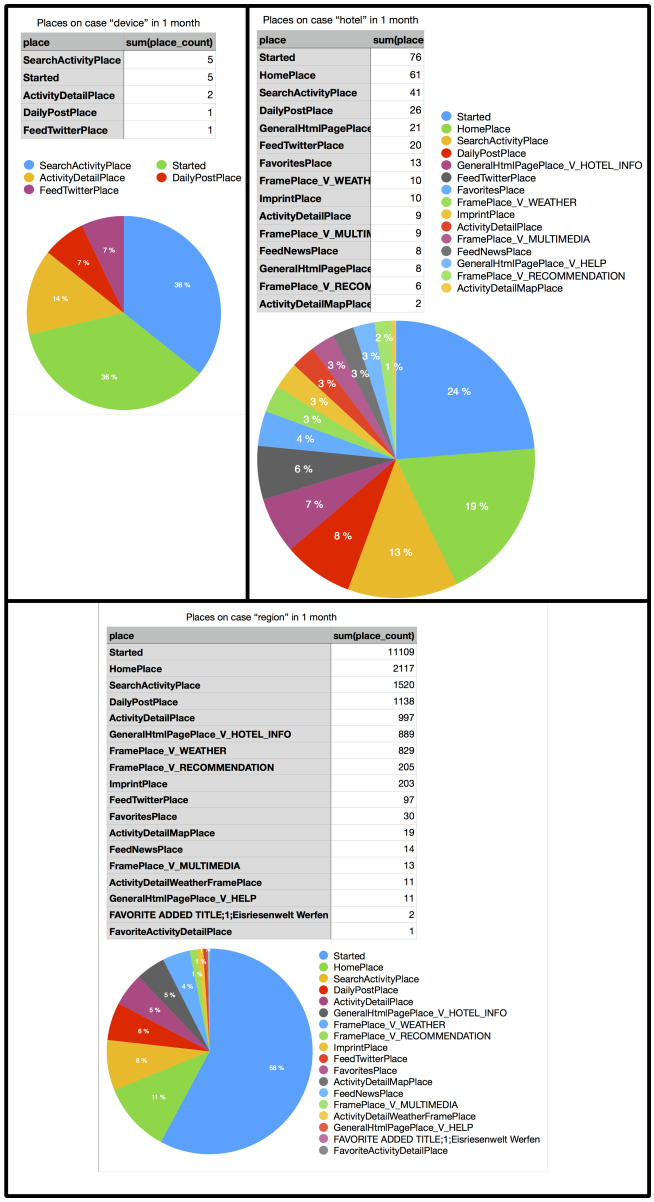
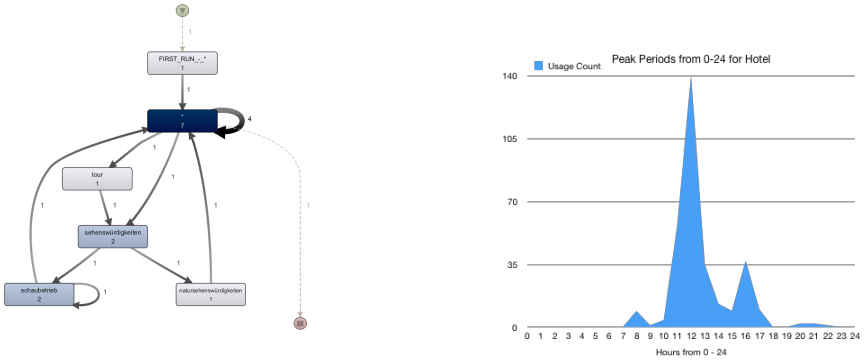


Fig. 7: Different cases for the most used services in *oHA*.

For storing logs, we would always prefer a relational database over a file system. It is much easier to deal with outliers or excluding test data on productive instances. Log modifications become easier and less time consuming as well. Also, transferring the logs from the data storage into a process mining tool



(a) Search Process (b) Peak Periods

Fig. 8: Analysis Results on Search Processes and Peak Periods

can be done faster now. The database in *oHA* is realized by a service running on a (web-)server, which is responsible for managing the data and receiving the logs from different systems. Doing so we have achieved a loose coupling between different systems and instances, which are reporting to a central data warehouse. The decision to separate the data warehouse server physically from the log generating applications bears advantages with respect to security, because there is only one server to protect. This is particularly challenging with respect to data from user applications where different regulations for different countries exist. Another recommendation is to create a scaleable architecture to be prepared for answering further questions and to integrate new systems. It was also useful to design non-time-critical micro services in the data warehouse for enriching and processing the stored log data, e.g., for mining and visualizing search processes. Finally, automating the log processing task reduces the failure rate with respect to conclusions on the guest behavior. Apart from technical aspects we recommend to identify relevant cases and to define analysis questions before starting to mine processes. The more cases are identified, the more expressive the questions can be as most questions can be asked from different viewpoints, e.g., for a region, a hotel, a device, or a user session.

One future goal in *CustPro* refers to improving the quality of the mined models based on semantic technologies in terms of, e.g., complexity. We also want to study the transferability to other industries.

References

1. van der Aalst, W., et al.: Process mining manifesto. pp. 169–194. Springer (2012)
2. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
3. Bose, R., Mans, R., van der Aalst, W.: Wanna improve process mining results? In: Computational Intelligence and Data Mining. pp. 127–134. IEEE (2013)

4. Eder, J., Olivotto, G., Gruber, W.: A data warehouse for workflow logs. Engineering and Deployment of Cooperative Information Systems pp. 117–121 (2002)
5. Gupta, G.: Introduction to data mining with case studies. PHI Learning Pvt. Ltd. (2014)
6. de Murillas, E.G.L., van der Aalst, W.M., Reijers, H.A.: Process mining on databases: Unearthing historical data from redo logs. In: International Conference on Business Process Management. pp. 367–385. Springer (2015)
7. Nabli, A., Bouaziz, S., Yangui, R., Gargouri, F.: Two-etl phases for data warehouse creation: Design and implementation. In: East European Conference on Advances in Databases and Information Systems. pp. 138–150. Springer (2015)
8. Ribeiro, J.T.S., Weijters, A.J.M.M.: Event cube: Another perspective on business processes. In: On the Move to Meaningful Internet Systems. pp. 274–283 (2011)
9. San Pedro Martín, J.d., Carmona Vargas, J., Cortadella Fortuny, J.: Log-based simplification of process models. In: Business Process Management: 13th International Conference, BPM 2015, Innsbruck, Austria, August 31-September 3, 2015: proceedings. pp. 457–474. Springer (2015)
10. Schamp, E.E.E., Schamp, E.: Status quo of big data analysis in small and medium size enterprises in Austria
11. Singh, A., Umesh, N.: Implementing log based security in data warehouse. International Journal of Advanced Computer Research 3(1) (2013)
12. Stolba, N.: Towards a sustainable data warehouse approach for evidence-based healthcare. na (2007)