Mining the Usability of Business Process Modeling Tools: Concept and Case Study

Tom Thaler[1], Dirk Maurer[2], Vittorio De Angelis[2], Peter Fettke[1], Peter Loos[1]

[1]Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany
{tom.thaler,peter.fettke,peter.loos}@iwi.dfki.de
[2]Software AG, ARIS Development, Saarbrücken, Germany
{dirk.maurer,vittorio.deangelis}@softwareag.com

Abstract. Business process models are key artifacts in business process management. The technical support of the process of process modeling is important for the quality and the applicability of the resulting models. The quality of that technical support plays an important role in the selection of corresponding software products and is a crucial characteristic of differentiation. Nevertheless, only little knowledge on the tool-specific line of actions and the corresponding challenges in the daily work of modelers is available, which makes it hard to improve a modeling tool against customer requirements. In order to address that conflict, we develop a method based on process mining, allowing the continuous analysis of modeling tools and the applied processes of process modeling with regard to software usability aspects. The resulting method containing the phases *user monitoring, trace clustering, usage model derivation, usage model analysis, recommendation derivation* and *implementation* primarily aims at a target-oriented design and further development of business process modeling tools and is evaluated with the ARIS Designer by performing a user study. The results allow promising estimations for an application of the method in a broader context.

Keywords: Business Process Modeling, Process Mining, Business Process Management, Lifecycle, BPM Use Case, ARIS

1 Introduction

Business process models are key artifacts in business process management. Traditionally, process models are generated by human modeling experts using modeling tools like the ARIS Designer of the Software AG. However, the process of process modeling still has many facets to discover. In fact, there is already manifold research on the process of process modeling like [1] describing a formal concept in order to study the modelers' sequence of actions or [2] investigating different modeling styles. However, the corresponding papers focus on general procedures of modeling, while the adequacy and the satisfaction of the technical support, which should be delivered with a particular process modeling tool, is not considered. With regard to such usability aspects, that makes it hard to improve a modeling tool against end user needs. In the past, the Software AG applied expert interviews, pre-release usability tests with pilot users of the ARIS community and other established usability methods in order to improve their modeling tool. However, only little knowledge on the challenges for the modelers' daily work could be identified and explicated.

Against that background, in the context of a research project we were looking for new approaches taking the real user behavior into account to be able to improve the modeling tool based on the real and not yet identified customer needs. Hence, the paper at hand aims at developing a method analyzing different dimensions of usability, whereby both the system design (in terms of a technical support of the process of process modeling) and the process of process modeling itself are explicitly addressed. Established process mining techniques are used for an automatic derivation of usage models which can then be enriched by manifold data like GUI information (e.g. element positions), or arbitrary user, system or context data (e.g. user experience). This renders it possible to analyze the real user behavior in detail and allows the target-oriented improvement of the modeling process and the corresponding software design, especially in terms of its usability. Referring to the general definition of software usability [3], the term "business process usability" should, in this paper, be understood as the extent to which a BPM software can be used for the effective, efficient and satisfactory management of business processes.

Since the potential method needs to combine different research fields from two different research disciplines, information systems (especially process mining) and software engineering (especially human computer interaction and usability engineering), it is necessary to identify the relevant literature from all fields involved. The identified methods and techniques are analyzed with respect to their applicability in the context of mining business process usability, which results in a collection of partial solutions for specific problems and a collection of gaps. To fill these gaps, a design science research approach is applied [4]. The approach of process mining is adapted with regard to the specific requirements of usability engineering. A phase model was developed and a corresponding tool support was implemented within the research prototype RefMod-Miner[1]. The resulting method is then evaluated in the context of a modeling scenario with the ARIS Designer, Version 9, of the Software AG by performing a user study.

After this introduction, Section 2 gives an overview of the related work in the mentioned research fields. Section 3 describes the developed method in the form of a continuous lifecycle, which is then applied in the context of the case study in Section 4. The results and limitations as well as some possibilities for transferring the method to other domains and application scenarios are discussed in Section 5. Section 6 gives an outlook on future developments and concludes this work.

[1] http://refmod-miner.dfki.de

2 Related Work

As mentioned above, in the context at hand, the research field of usability engineering as well as information system research in general and process mining in particular are of major importance.

Traditional methods measuring the usability of software are affected by manual and empirical approaches often applying questionnaires, user or expert interviews and observations [5]. In fact, methods like eye-tracking or click-path-analyses are partially automatable and literature concerning the application and design of such controlled experiments in terms of using adequate test methods with a meaningful configuration (e.g. sample size) is available [6]. However, the environment settings (e.g. laboratory) as well as the data analyses are very expensive in most cases.

Above all, there are some works using log data as a basis for the automated analysis of software usability (e.g. [7, 8, 9]), which also contain approaches using event patterns to measure specific aspects of usability (e.g. [10, 11]). Isolated works [12, 13] also derive process models (petri-nets) and address some possibilities of usability analysis. However, the used mining methods are rather rudimentary as they do not take today's aspects of process mining, like dealing with noise or a harmonization of log data of different systems, into account. In fact, the used mining techniques cover the beginning of the process mining era, e.g. [14, 15], and a further consideration of current methods and techniques from the information systems research is missing.

Nevertheless, process-orientation is a core characteristic of business (process management) software supporting concrete business tasks and processes in a technical manner. Therefore, the application scenario of usability is of high importance for the information systems research and for the design of business information systems in general. Current approaches, e.g. genetic algorithms [16] or cluster techniques handling noisy data or avoiding spaghetti-like models [17] could be helpful in that context. Not till then, it is possible to derive meaningful models or meta-information enabling researchers or practitioners to draw concrete conclusions with reference to usability aspects. Especially a combination of different process or model metrics from different research disciplines – like those from [18] or [19] in the context of business process analysis and from [20] or [7] in the context of usability engineering and usage analysis – might improve the evaluation of business processes and their implementation. That metrics makes it possible to quantify several aspects concerning the quality, the understandability and also the usability of business processes and their application in an automated manner. In contrast to existing methods of usability engineering, there are also further application scenarios like the automatic derivation and further development of software reference models in general and usage models in particular [21].

Indeed, both research fields address similar approaches towards an automatic derivation of process models but the current states of research strongly diverge. A transfer, adaption and further development of both states of research will imply an enrichment of the respectively other discipline.

3 Generic Concept Development

For the design of an adequate method and tool support, the approach of process mining was adapted with regard to the specific requirements of usability engineering. A phase model (Fig. 1) containing the phases *user monitoring*, *trace clustering*, *usage model derivation*, *usage model analysis*, *recommendation derivation* and *implementation* was developed and serves as a continuous lifecycle [22]. In contrast to previous approaches, that lifecycle allows an application in the live operation and, thus, takes the real user behavior into account. Therefore, mostly expensive laboratory experiments are not necessary.

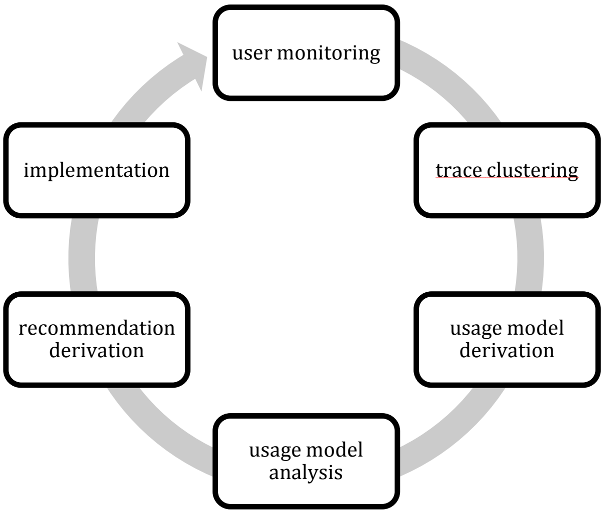

Fig. 1. Business Process Usability Mining Lifecycle

3.1 User Monitoring

Process execution data (instance data) are the basis for business process usability mining. Depending on the analysis objectives, there are different requirements for log data. Generally, it is necessary to fulfill the traditional log data requirements of process mining (case, task, originator, timestamp) [23], which should then be extended with additional information depending on the context. In case of an identification of usability weak points, it might be helpful to log e.g. GUI information like element positions or case-specific data. Collecting further information may imply the use of further data sources like an enterprise database, external services or sensors. Since software-as-a-service plays an increasing role in the business context, (web) server logs or error logs, which are traditionally not considered in the context of process mining, are possible as well. Against that background, one needs to design a logging strategy based on the analysis objectives or the application scenario and implement it in the addressed software. Furthermore, a consolidation of different data sources is of high importance.

3.2 Trace Clustering

Trace clustering describes the task of clustering traces within log data concerning a specific cluster criterion and is a traditional challenge in the context of process mining. As, in general, business software as well as BPM tools cover a multitude of different business processes, a corresponding log file covers all these processes, too. Discovering a process model based on a non-clustered log file leads to a highly complex and non-human-readable models in most cases (so-called spaghetti-like models). This makes it necessary to identify different processes or instance classes in order to generate several process models with less complexity or similar characteristics (e.g. [24, 25]). The choice of a particular trace clustering technique thereby highly depends on the analysis objectives [26]. Thus, there are manifold aspects, which may serve as a criterion for trace clustering as e.g.:

- processes: e.g. variants, patterns, occurrence of loops or tasks
- resources/performance: e.g. time, budget, hardware, load values
- cases: e.g. value of a shopping cart
- users: e.g. experience, age, groups
- software: e.g. version, device

Thus, the recorded log data can be interpreted as a multidimensional data cube, whose dimensions are partially not known a priori (see Fig. 3). In fact, some dimensions are given by the log specification (the recorded attributes), others, like the actually recorded process, are unknown. Against that background, it is partially possible to apply slicing and dicing approaches from the area of data warehouses. However, especially in the context of business process usability analysis, the identification of new information and patterns regarding the proceeded processes and variants, possibly depending on user profiles, are of major importance. A good overview of existing trace clustering techniques with its corresponding implementations and an evaluation of their applicability in different contexts is presented in [26].

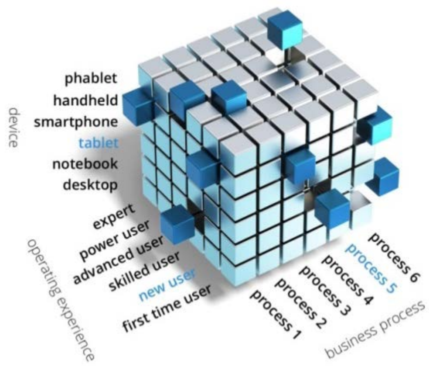

Fig. 2. Illustration of cluster dimensions as a data cube

3.3 Usage Model Derivation

Process mining distinguishes three different fields: (1) process discovery, (2) conformance checking and (3) enhancement [17]. Process discovery aims at deriving a new process model solely based on log data, while conformance checking addresses the comparison of the as-is process to the to-be process. Enhancement focuses the derivation of new information from log data and annotating it to an existing process model.

Against that background, all of the mentioned fields play an important role for business process usability mining in general and in the phase of usage model derivation in particular. In that phase, one needs to derive a process model based on the clustered log data and to enrich it with further information like performance data, execution probabilities, correlation matrices and further (scenario-specific) data and metrics. Today, a lot of different process mining techniques with different characteristics of the output models do exist already. They differ in the fitness and appropriateness of the resulting models to the underlying log files, e.g. in their simplicity, in their abstraction level, in the resulting model type (petri-nets, EPC, FSM, etc.) [27] or in their calculatory approach. Thus, a concrete algorithm should, again, be selected depending on the concrete analysis objectives [e.g. 14, 16, 28, 29]. In contrast to discovery and enhancement, conformance checking should be seen in the phase of usage model analysis (phase 4) as, especially in the context of business process usability mining, it might be of major importance to know whether the users utilize a software in the intended way.

3.4 Usage Model Analysis

There are several possibilities of analyzing the usage model. First of all, many metrics from different research fields exist and are able to characterize the model(s) and give first indications to particular weak points:

- model metrics: e.g. complexity, extent, cross-connectivity [18, 19]
- process metrics: e.g. execution count, execution time, error rates, cancellation rates
- usability metrics: e.g. irrelevant actions, undo actions, using help function

These categories can also be broken down into further subcategories, e.g. size and complexity in terms of model metrics or placement and time aspects in terms of usability metrics. Disregarding these metrics, there are several further aspects, e.g.:

- Achievement of objectives / conformance checking: In the context of business processes and their management, oftentimes, there are objectives which should be achieved at process executions. These could be the overall execution time of a process, the consumption of resources, etc. Also business rules which are obligatory at the process execution, e.g. coming from legal aspects, might be important for the determination of conformance.
- Causal dependencies: Process models may contain causal dependencies between activities or process fragments, which are not evident in the process model. A correlation matrix may uncover those dependencies.

- Core and exception fragments: Oftentimes, process models contain activities or fragments which are executed in a high amount of cases (core actions) as well as those which are executed very seldom (exception actions). Knowledge about that frequency helps focusing on the most important system points during development.
- Non-supported processes: Sometimes users use a system for the execution of processes which are not intended by the system producer. Identifying these processes helps improving a system against customer needs or may help identifying further business areas.
- System avoidance: Apart from the use of a system for non-supported processes, users also avoid systems at executing particular process steps. Avoiding a functionality although it is available may be an indication of a non-working or badly implemented functionality.

In a nutshell, simple statistical indicators might lead to a first hint concerning process or software usability issues but are not able to analyze these issues in detail. In most cases, further information on the process and its execution logs is needed, which bases on input from human experts.

Since it is not possible to calculate the mentioned metrics from the different areas on scratch, we implemented a tool support in the research prototype RefMod-Miner. An extract of the available metrics and statistical analysis techniques are illustrated in the screenshots presented in Fig. 5. Furthermore, it is necessary to have additional functionalities allowing the graphical navigation through the model, like the visualization of predecessor and successor nodes of a specific node or the highlighting of particular nodes (like help or undo calls). The application of these analysis techniques may lead to concrete hints to weak points in the technical support of modeling.

3.5 Recommendation Derivation

The recommendation derivation phase aims at interpreting the usage model analysis results and delivering concrete hints concerning the business process and software usability improvement and the further development of the system according to the real customer needs. The recommendation should help the system producer answer e.g. the following questions:

- Are there weak points in the system concerning the process support (e.g. avoided functionalities, needless undo-actions within process execution, misuse of functionalities, missing functionalities, unclear labels, very seldom or unsed buttons/functions at prominent positions, long loading times)?
- What are the core application scenarios at the user side? / Which implemented processes or functionalities are not used?
- Are there observable user profiles apart from user role or experience? Are there significant differences in using a system?

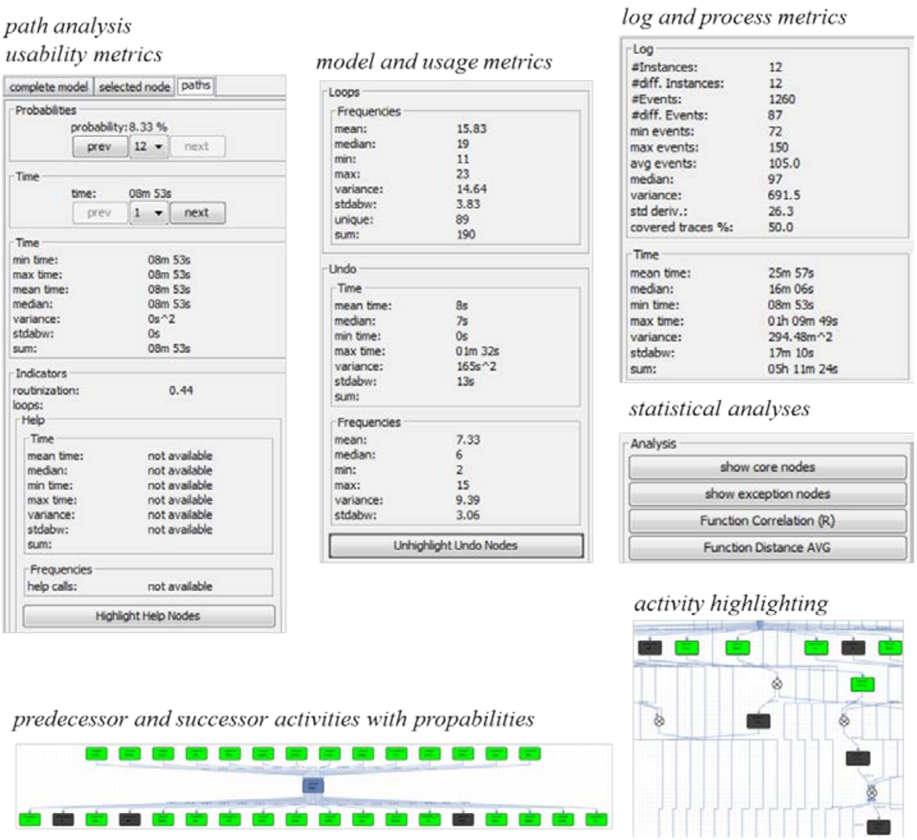

Fig. 3. Extract of metrics and analyses tools implemented in the RefMod-Miner

- Are there observable case profiles for a process influencing its execution?
- Are there further functional requirements at the user side?
- Are there possibilities to improve the process (e.g. user- or case-sensitive processing, adding new functionalities, data preloading, reorganization of forms)?

These questions should be answered based on the usage model analysis results. In fact the results deliver hints to (potential) critical points of the usage models. However, there are currently no adequate technologies making it possible to answer these questions in an automated manner. Instead, expert knowledge is needed to interpret the analysis results and to derive concrete improvement capabilities.

3.6 Implementation

The implementation phase covers the selection, design, planning and implementation of possible solutions for the generated hints and solution capabilities. Thus, this phase acts as a completion of a particular lifecycle iteration and leads to a new software release. At the same time, it marks the beginning of a new lifecycle iteration

4 Case Study

4.1 Design and Setup

The presented lifecycle approach should now be instantiated and evaluated by applying it to a real world scenario. Against that background, we performed a user study in a modeling scenario. 13 students needed to (1) model an organigram, (2) model an EPC and (3) modify an EPC using the rich modeling client of the ARIS Designer of the Software AG based on natural language text descriptions. The focus of the study was to gain knowledge on how users act to reach a solution, not the correctness of the produced solution itself. Against that background, the interactions of the users were tracked in a specific way, which is described below. 10 of the students already had modeling experience, while the other 3 had not. The average time needed for executing the tasks was 47 minutes. In order to be able to validate concrete business process usability issues derived from the proceeded analyses, additionally, the user screen was recorded by a screen capturing software.

4.2 Application and Findings

User Monitoring. As a first step and prior to the actual user study it was necessary to develop an adequate logging strategy collecting the relevant information to be able to generate detailed knowledge on how the users interact with the system in the context of process modeling. Thus, in addition to the traditional log information of process mining (caseid, task, timestamp and initiator) two further attributes, namely the callitem describing in which way the user triggers particular actions, e.g. with the mouse, using a shortcut, using an item of the symbol bar, etc., and the involved objects as e.g. the elements on the grid are recorded as well. A sample log file generated by the ARIS 9 Designer in the user monitoring phase is presented in Fig. 2.

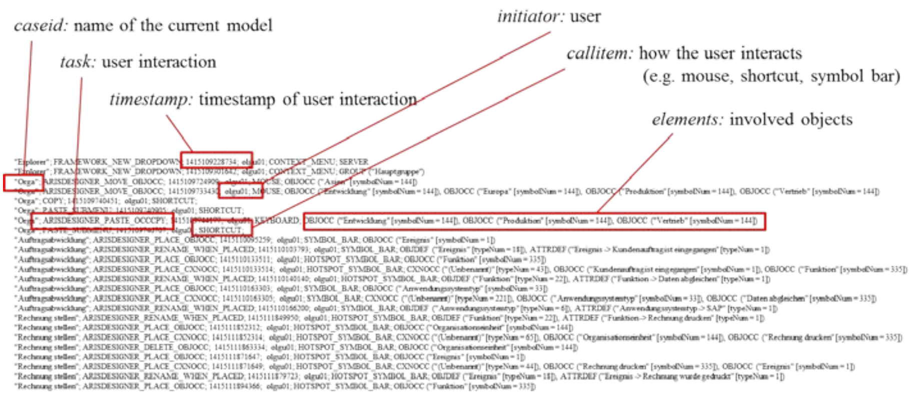

Fig. 4. Sample log file generated in the user monitoring phase

Trace Clustering. Since the user study contains three different exercises, in the trace clustering phase we initially focused on separating the log file based on the processes which are equivalent to that exercise. Prior to the clustering, we knew that the each of the exercises contained actions which are in all probability not proceeded in the two others. Referring to [26], we used that known causal dependency as a basis for separating the log file. As a second direction, we separated the log file based on the information of whether a user has experiences in working with ARIS or not.

Usage Model Derivation. In order to derive the usage models based on the clustered log files, we applied the Heuristics Miner [29] with default parameters. Thereby, the log files were additionally prepared in two different settings: (1) task, callitem and elements are consolidated and use a task description, e.g. "place event 'Event' by HOTSPOT_SYMBOL_BAR" and (2) the task is used as it is recorded, e.g. "place event". Extracts of the resulting EPC models from the two settings are presented in Fig. 4. One can easily see, that the complexity of the models differs in a high degree, which is grounded in the fact that setting 1 produces much more detailed node labels and, thus, a significantly higher absolute amount of nodes than setting 2. Hence, the degree of the proceeded task description consolidation again depends on the analysis objectives.

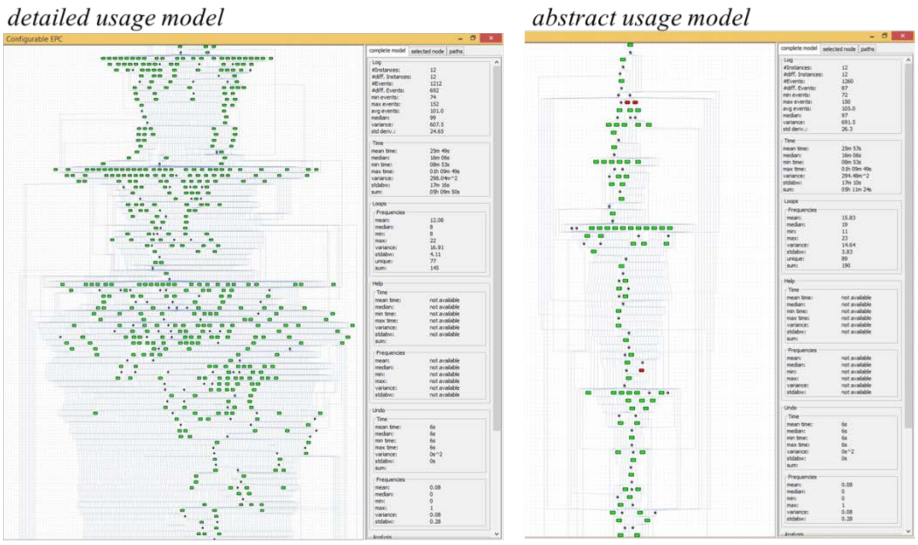

Fig. 5. Detailed vs. abstract usage model visualized with the RefMod-Miner

Usage Model Analysis. Analyzing and interpreting these models by applying the analysis functionalities of the research prototype RefMod-Miner led to the identification of manifold aspects, ranging from a purely technical to a professional perspective. Three of them are exemplarily presented in Fig. 6.

The first example shows that users were not able to understand the toggled-edge-mode. When activated, they expected an automatic connection of edges based on the element positions which led to the effect that modelers e.g. placed connectors over the edges connecting several nodes. In contrast to their expectations, the connector was not automatically connected.

The second case covers a more professional aspect. Some modelers placed organizational units to the grid and connected them to an activity. They expected the connecting edge to be undirected, however, the system automatically produced a directed edge from the organizational unit to the activity. The only solution to the arisen problem was the manual deletion of the edge direction for all corresponding edges. A similar case showed that it was not possible to modify, respectively change, the edge direction, which might be meaningful in many contexts.

In contrast to that, the third case uncovers different strategies in modeling. While some modelers placed the nodes and labeled them immediately, others primarily added a set of nodes and labeled all of them afterwards, which needs much more time.

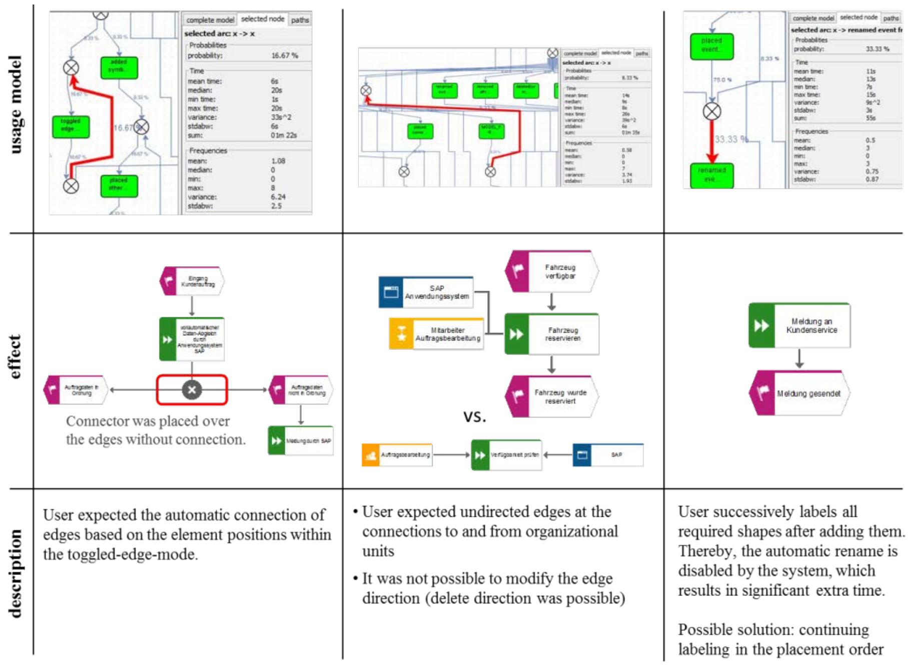

Fig. 6. Presentation of three identified issues

Recommendation Derivation. The first issue results from the fact, that users were not able to interpret the meaning of the “toggled-edge-mode”. Thus, renaming the functionality might improve the understandability of it. In contrast to that, the second issue constitutes a bug, which can simply be fixed by allowing the modification of edge directions. Since the third issue uncovers a user demand (respectively a not yet considered modeling strategy), it is necessary to provide a new functionality supporting that strategy. A continuous labeling in the placement order might be a meaningful feature for that specific demand.

Implementation. For some of the identified issues, improvement potentials were developed and already planned for the implementation phase of the next software release, which is currently worked on.

5 Discussion

Despite the early stage of studying and applying the approach in a real context and although the number of participants in the user study was small, as well as the considered scope in the software, it was possible to identify 10 different issues ranging from minor bugs and general weak points to specific user demands. Also the derived information were detailed enough to be able to describe them in a professional way and to address them with concrete improvements which are currently being implemented. This shows the promising potentials for an application of the approach in a broader user study with pilot users and beyond.

However, there are also some aspects in the context of the case study, which should be discussed. Although the achieved results are promising, the statistical relevance of the particular identified issues and user demands is currently unknown. In the context of a further study with more participants, it would be necessary to determine the statistical relevance with statistical tests, e.g. using the p-Value.

From a technical point of view, the possible amount of upcoming data will require the use of methods which are able to handle it. Depending on the degree of detail of the log files (e.g. every click, every mouse movement, etc.), on the number of monitored users and on their intensity of use, the log files will become very memory-intensive. Thus, their content will become more complex, as the proceeded case study has already elucidated. This leads to several challenges as e.g. the user tracking, the clustering of the log files and a potentially high complexity of resulting usage models, which are hard to interpret by human experts. However, first methods and techniques addressing these challenges do already exist and need to be evaluated with regard to their applicability in the context at hand.

Generally, one might ask the question of whether it is expedient to improve the usability of software tools supporting the process of process modeling as opposed to training the end users. In fact, an adequate training might help the users to work more efficiently, effectively and satisfactory with a modeling tool, however, individual approaches of process modeling are important as well. Against that background, it is necessary to do both, train the end users and improve the usability based on the end users' needs. Additionally, it should be evaluated whether the user training or the software improvement leads to more promising results regarding business process usability, e.g. in terms of efficiency, effectiveness, satisfaction and also costs, which is an important factor especially for small and medium enterprises. We assume that the cost factor is a particular major strength of the developed method.

With regard to the transferability of the developed method to other domains and applications, we identified manifold promising potentials. Thus, not only in the context of process modeling, also in the context of business process supporting software in general, as e.g. ERP or workflow systems, an application of the developed method might be promising in different scenarios. In addition to the already mentioned objectives, the identification of end user needs and the analysis of the business process usability, we identified, amongst others, the following application scenarios [22].

- Controlling the Software Evolution. Further development of a software is in the nature of a product lifecycle. Nevertheless, it is challenging to evaluate whether a further development leads to the desired effect and whether it is used as it is intended. This affects both new supported processes and adapted existing processes. Since the developed method analyzes the real user behavior, it is possible to follow the software evolution from the user side. This can also be seen in [30].
- Inductive Usage Reference Model Development. Information about the process performance, the resource consumption and other collected data allow the inductive development of reference models with best practice (as known) character. Based on the process instances, a process model could be derived concerning different objectives like the minimization of cost or resources or the optimization of the output quality.
- Ease of Learn. One quality criterion of a business process supporting software might be the effort necessary to be able to operate it in an adequate manner. An analysis of the usage models of users over time would visualize their learning effects and, thus, allow the derivation of individual learning curves.

6 Conclusion and Outlook

The paper at hand presents a method for mining the usability of business process modeling tools based on process mining. It constitutes different aspects, which need to be investigated in order to be able to gather hints on the further development of a corresponding software according to the real customer needs. While the phases of user monitoring, trace clustering and usage model derivation already have an established theoretical and technical foundation which can be adapted concerning usability aspects, a detailed analysis of the resulting data seems to be challenging. In fact, there are several ideas quantifying the usability of a software system and characterizing process models. However, these ideas need to be further developed, conceptualized, implemented and evaluated.

We were able to show that the developed method creates the missing link between the software engineering view and the process-oriented view on business process supporting software. This leads to promising potentials for their design and further development. Moreover, several promising scenarios for a meaningful application of the method in other domains could be identified and will be addressed in future work.

Finally, the developed method has several advantages over existing approaches. It can be applied in production use and in real environments and, thus, involves the real user behavior. At the same time, it obviates a deformation of measurement results, which traditionally constitutes a problem of direct observations. Moreover, the measurement and analysis of usability aspects can, in many cases, be arranged automatically or with only little input, which leads to significantly lower costs and, thus, also enables small and medium enterprises to apply the method.

References

1. Pinggera, J., Zugal, S., Weidlich, M, Fahland, D., Weber, B., Mendling, J. Reijers, H.: Tracing the Process of Process Modeling with Modeling Phase Diagrams, In: Proceeding of the Business Process Management Workshops 2012, pp. 370–382, 2012.
2. Pinggera, J.,Soffer, P., Zugal, S., Weber, B., Weidlich, M., Fahland, D., Reijers, H., Mendling, J.: Modeling Styles in Business Process Modeling. In: Enterprise, Business-Process and Information Systems Modeling, pp. 151–166, 2012.
3. ISO: 9241: Ergonomic requirements for office work with visual display terminals. Part 11: Guidance on usability, 1998.
4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 2004, pp. 75-105.
5. Nielsen, J.: Usability Engineering. Academic Press, Boston, 1993.
6. Kohavi, R., Longbotham, R., Sommerfield, D., Henne, R. M.: Controlled experiments on the web: survey and practical guide, In: Data Mining and Knowledge Discovery, Vol. 18, Issue 1, pp. 140-181, 2009.
7. Ivory, Y., Hearst, M.A.: The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys 33, 2001, pp. 470-516.
8. Seffah, A., Donyaee, M., Kline, R.B., Padda, H.K.: Usability measurement and metrics: A consolidated model. Software Quality Control 14, 2006, pp. 159-178.
9. Ting, I., Kimble, C., D., K.: UBB Mining: Finding Unexpected Browsing Behaviour in Clickstream Data to Improve a WebSite's Design. In: IEEE/WIC/ACM International Conference on Web Intelligence, 2005, pp. 179-185.
10. Hornbaeck, K.: Current practice in measuring usability: Challenges to usability studies and research. International Journal of Human-Computer Studies 64, 2006.
11. Shah, I.: Event Patterns as Indicators of Usability Problems. Journal of King Saud University – Computer and Information Sciences 20, 2008, pp. 31-43.
12. Hilber, D. M., Redmiles, D. F.: Extracting Usability Information from User Interface Events. ACM Computing Surveys (CSUR) 32, 1999, pp. 384-421.
13. Siochi, A. C., Ehrich, R. W.: Computer analysis of user interfaces based on repetition in transcripts of user sessions. ACM Trans. Inf. Syst. 9, 1991, pp. 309-335.
14. van der Aalst, W. M. P., Weijters, A. J. M. M., Maruster, L.: Workflow Mining: Discovering process models from event logs, 2005.
15. Cook, J. E., Wolf, A. L.: Discovering Models of Software Processes from Event-Based Data. ACM Transcations on Software Engineering and Methodology 7, 1998.
16. Ana Karla, M.: Genetic Process Mining. CIP-Data Library Technische Universität Eindhoven, Eindhoven, 2006.
17. van der Aalst, W.: Process Mining: Discovery, Conformance and Enchancement of Business Processes. Springer, Berlin, Heidelberg, 2011.
18. Melcher, J.: Process Measurement in Business Process Management – Theoretical Framework and Analysis of Several Aspects. KIT Scientific Publishing, Karlsruhe, 2012.
19. Mendling, J., Sánchez-González, L., García, F., La Rosa, M.: Thresholds for error probability measures of business process models, In: Journal of Systems and Software, Issue 85, Volume 2, pp. 1188-1197, 2012).

20. Balbo, S., Goschnick, S., Tong, D., Paris, C.: Leading Usability Evaluations to WAUTER. In: 11th Australian World Wide Web Conference, 2005.
21. Kassem, G.: Application Usage Mining: Grundlagen und Verfahren. Shaker Verlag, Aachen, 2007.
22. Thaler, T.: Towards Usability Mining, In: Lecture Notes in Informatics. Jahrestagung der Gesellschaft für Informatik (INFORMATIK-14), Big Data – Komplexität meistern, September 22-26, Bonner Köllen Verlag, Stuttgart, Germany, 2014.
23. van der Aalst, W. M. P.: Decision Support Based on Process Mining. In: Burstein, F., Holsapple, C. W. (eds.) Handbook on Decision Support Systems 1. Basic Themes, Springer, Berlin, Heidelberg, 2008, pp. 637-657.
24. Ekanayake, C. C., Dumas, M., García-Bañuelos, L., La Rosa, M.: Slice, Mine and Dice: Complexity-Aware Automated Discovery of Business Process Models. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, LNCS 8094, Springer, Berlin, 2013, pp. 49-64.
25. Jagadeesh, R. P., van der Aalst, W. M. P.: Trace alignment in process mining: opportunities for process diagnostics. Proceedings of the 8th international conference on Business process management, Springer-Verlag, Hoboken, NJ, USA, 2010, pp. 227-242.
26. Thaler, T.; Ternis, S. F.; Fettke, P.; Loos, P.: A Comparative Analysis of Process Instance Cluster Techniques, In: Proceedings of the 12th International Conference on Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik (WI-15), March 3-5, Osnabrück, Germany, Universität Osnabrück, 3/2015.
27. Thaler, T.: Entwicklung einer Methode zum Process Mining unter besonderer Berücksichtigung von Organisationswissen. Semiramis Research and Service Unit, 2013.
28. Weijters, A. J. M. M., van der Aalst, W. M. P., de Medeiros, A. K. A.: Process Mining with the Heuristics Miner-algorithm. BETA Working Paper Series WP 166, 2006.
29. Weijters, A. J. M. M., Ribeiro, J. T. S.: Flexible Heuristics Miner (FHM). In: IEEE Symposium on Computational Intelligence and Data Mining, 2011.
30. Thaler, T., Fettke, P., Loos, P.: Process Mining - Fallstudie leginda.de. HMD Praxis der Wirtschaftsinformatik 293, 2013, pp. 56-66.