Towards Usability Mining

Tom Thaler

Institute for Information Systems (IWi) at the
German Research Center for Artificial Intelligence (DFKI) and Saarland University
Campus, 66123 Saarbrücken
tom.thaler@iwi.dfki.de

Abstract: The usability of business information systems (BIS) plays an important role in the selection of corresponding software products and is therefore a crucial characteristic of differentiation. Nevertheless, there are no automatic techniques considering the process aspects of these systems in an adequate manner. Hence, the paper at hand aims at the automatic derivation of software usage reference models, which serve as a basis for measuring the usability of business information systems and its supported process. Apart from conventional event logs, further data, like element positions or sensor data, should also be taken into account. Analyzing these data in the context of business process usability and software usability bases on established techniques from the fields of business process analysis and usability engineering. The resulting method "Usability Mining" aims at the target-oriented design of BIS and an adequate technical support for business processes.

1 Introduction

The usability of business information systems (e.g. ERP, workflow, systems or even particular business software as well as production control systems) plays an important role in the selection of corresponding software products. It is defined as the extent to which a software product can be used for the effective, efficient and satisfactory achievement of business objectives by a user [IS98]. Thereby, business information systems have special characteristics in terms of process orientation since they intend the technical support of concrete business processes. Hence, usability should, in that context, be understood as the extent to which a business information system can be used for the effective, efficient and satisfactory execution of business processes.

Indeed, the supported business processes deliver a basic structure for the software and user interface design, but, at the same time, the practical application can lead to inefficiency, ineffectiveness or disaffection, e.g. in terms of stress. The reasons for that can be quite divergent. Missing or bad positioned elements as well as long loading times make a process more time consuming than necessary and lead to uncertainty and impatience at the user side. A high process complexity impedes the process execution and negatively affects the process efficiency.

Against that background, the paper at hand aims at developing a method analyzing different dimensions of usability, whereby both, the system design and the process design, are explicitly addressed. Established process mining techniques are used for the automatical derivation of usage models which can then be enriched by manifold data like GUI information (e.g. element positions), sensor data (e.g. process data from cyber-physical systems or health data like pulse or body temperature) or additional user data (e.g. user experience). This renders it possible to analyze the real user behavior in detail and allows the target-oriented improvement of process and software design. The whole method is called "Usability Mining" and allows the consideration of different application scenarios, such as- the following:

Identification of Customer Needs. Sometimes users apply a software for processes or activities, which are not intended by the producer. Mining usage models of a software helps to identify these processes or activities and allows the further development of the software with respect to the actual customer needs (e.g. [TPL13]).

Measurement of Software and Process Usability. Software and process usability are closely connected to each other as the design of a process directly affects the implementation and vice versa. Usage models and their analysis, e.g. in terms of specific metrics, make it possible to assess the usability of processes and their implementation and can lead to target-oriented optimizations.

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 usability mining analyzes the real user behavior, it is possible to follow the software evolution from the user side. This can also be seen in [TPL13].

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 minimization of cost or resources, reduction of stress or optimization of quality.

After this introduction, section 2 describes the research methodology. Section 3 gives an overview of the related work in the relevant research fields, while section 4 presents the usability mining lifecycle with a description of the different phases and an attendant running example. Based on that lifecycle, section 5 concludes the need for further development and summarizes some particular areas of work.

2 Methodology

As the method combines 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 relevant literature from all of the involved fields. After that, a phase model containing the different steps of usability mining (depending on the application scenario and based on the procedure of process mining) is elaborated. The identified methods and techniques will be analyzed with respect to their applicability in the context of usability mining which results in a collection of partial solutions for specific problems and a collection of gaps. In order to fill these gaps, a design science research approach is applied [HMP04]. Thus, it is planned to apply an integrated evaluation in the following real application scenarios:

- ARIS: The ARIS Designer is a modelling platform from Software AG which is available both as a native and a web application. It is planned to analyze the process of process modelling in the context of an experiment.
- MIRROR: The research prototype MIRROR (web application) covers organizational learning processes and is used in different hospitals. It is planned to analyze these organizational learning processes in the whole and over all involved hospitals.
- DFKI: The German Research Center for Artificial Intelligence hosts various organizational web-based software e.g. for time recording, acquisition or leave requests. It is planned to analyze the travel management processes in operation.

3 Related Work

As mentioned above, in the context at hand, especially 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, whereby, in most cases, questionnaires, user interviews and observations or expert interviews are applied [Ni93]. 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 [KLS09]. 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. [IH01, SDK06, TK05]), which also contain approaches using event patterns in order to measure specific aspects of usability (e.g. [Ho06, Sh08]). Isolated works [HR99, SE91] also derive process models (petri-nets) and address some possibilities of usability analysis. However, the used mining methods are rather rudimentary as, nowadays, aspects of process mining, like dealing with noise or a harmonization of log data of different systems, are not taken into account. In fact, the used mining techniques cover the beginning of the process mining era, e.g. [AW05, CW98], 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 information systems as products like ERP or workflow systems support 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. Current approaches, e.g. genetic algorithms [Ka06] or cluster techniques handling noisy data or avoiding spaghetti-like models [Aa11] 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 in reference to usability aspects. Especially a combination of different process or model metrics from different research disciplines – like those from [Me12] in the context of business process analysis and from [BGT05] or [IH01] in the context of usability engineering and usage analysis – might improve the evaluation of business processes and their implementation. 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 [Ka07].

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.

4 Usability Mining Lifecycle

In order to realize a concept for usability mining, the author developed a six phase lifecycle consisting of (1) user monitoring, (2) trace clustering, (3) usage model derivation, (4) usage model analysis, (5) recommendation derivation and (6) implementation. All phases are described in the following subsections with a description of input, content (possible techniques) and output. In order to exemplarily demonstrate the application of the lifecycle, a running example is being executed through the lifecycle. The example is based on a real business scenario of a translation service in the web, a detailed process description can be found in [TPL13].

4.1 User Monitoring

Process execution data (instance data) are the basis for usability mining. Depending on the analysis objectives, there are different requirements for the log data. If, for example, the intention is to develop a usage reference model in an inductive manner, it is satisfactory to fulfill the traditional log data requirements of process mining (case, task, originator, timestamp) [Aa08], while in case of an identification of usability weak points, it might be helpful to gather further 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 measuring vital parameters. Since software-as-a-service plays an increasing role in the business context, (web-)server logs or error logs are supposable as well, which are traditionally not considered in the context of process mining. 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.

Input: User interaction

Content: Logging execution data [Aa11], enhance log data with further information (e.g. from enterprise database) [JJ13] if necessary, (web-)server protocols

Output: Log data (event logs)

Running example: As the log data from the running example were already collected, there is no possibility of gathering further information afterwards. The structure corresponds to the traditional process mining requirements. Thus, there is information about the case, the task, the originator (anonymized) and the timestamp. In addition, there is also information about release changes. Thus, the log data are enriched by the system version (Table 1). Apart from the available data, in the context at hand, it would be meaningful to log some further data attributes like the language combination, the size of the customer's company, etc. In this specific scenario, these data would be of major interest in terms of the clustering of the log data.

Table 1: Sample log file of running example

case_id	task	originator	timestamp	system_ver
5812	create translation case	Customer	16.06.2011 09:01	2
5812	request calculation	Customer	16.06.2011 09:04	2
5812	offer calculation	Controller	16.06.2011 10:11	2
5812	cancel case	Controller	02.08.2011 09:16	3
5813	create translation case	Customer	16.06.2011 09:05	3
5813	order translation	Customer	16.06.2011 09:05	3
5813	offer translation	Controller	16.06.2011 09:13	3
5813	accept offer	Translator	16.06.2011 12:41	3
5813	do translation	Controller	17.06.2011 15:31	3
5813	deliver translation	Controller	17.06.2011 15:31	3

4.2 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 information systems 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 not human readable model 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. [EDG13, JA10]). Depending on the objectives of the application scenario, there are also further (additional) aspects, which may serve as a criterion for trace clustering:

- Users: e.g. experience, age, groups
- Software: e.g. version, device
- 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

Input: Log data (e.g. log file)

Content: Particular trace clustering techniques depending on the objective, e.g. [JA09, MGG08, SA08]

Output: Clustered log data

Running example: Since the log data of the running example contains several system version which differ in process, first of all, the log is clustered by the system versions.

4.3 Usage Model Derivation

Process mining distinguishes three different fields: (1) process discovery, (2) conformance checking and (3) enhancement [Aa11]. 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 them to an existing process model.

Against that background, all of the mentioned fields play an important role for 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 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 already exists. For example, they differ in the context of detailedness (to what extent do the resulting models cover the trace contained in the log file), abstraction level, model type (petri-nets, EPC, FSM, etc.) [Th13] or the calculational approach. Thus, a concrete algorithm should, again, be selected depending on the objectives. In contrast to discovery and enhancement, conformance checking should be seen in the phase of usage model analysis (phase 4).

Input: Clustered log data

Content: Particular process mining algorithms depending on the objective (e.g. [AW05, Ka06, WAM06, WR11])

Output: Process model

Running example: Using the Heuristics Miner [WAM06] with default parameters leads to the process model in Figure 1.

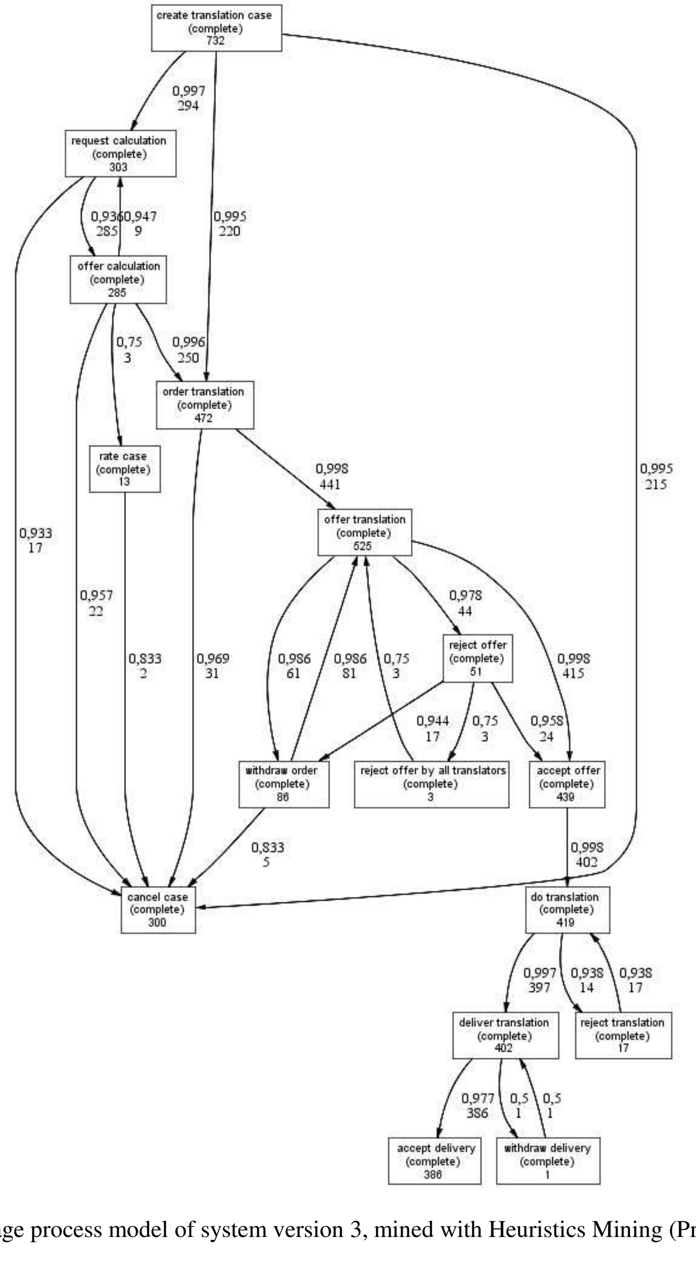

Figure 1: Usage process model of system version 3, mined with Heuristics Mining (ProM 5.1)

4.4 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 [Me12]
- 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, 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 obligatory at the process execution, e.g. coming from legal aspects, are possible to occur.
- 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 processes, for which it is 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 can sometimes 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 to the process and its execution logs is needed, which bases on the input from human experts.

Input: Process model

Content: Metrics, conformance checking techniques, correlation analysis, path analysis, behavioral analysis (based on instance tracking), etc.

Output: Metric values, conformance values, correlations, path frequencies, etc.

Running example: In order to analyze the usage / usability, the author implements a research prototype in the RefMod-Miner framework of the IWi at the DFKI. The currently implemented functionality calculates different metrics like element probabilities, duration of element and path executions as well as their frequencies with further statistical data (variance, standard derivation, median, etc.). In addition, the most or less frequent paths as well as the fastest or slowest process instances are being identified. That allows a more detailed analysis of relevant model fragments, also based on the analyst’s needs.

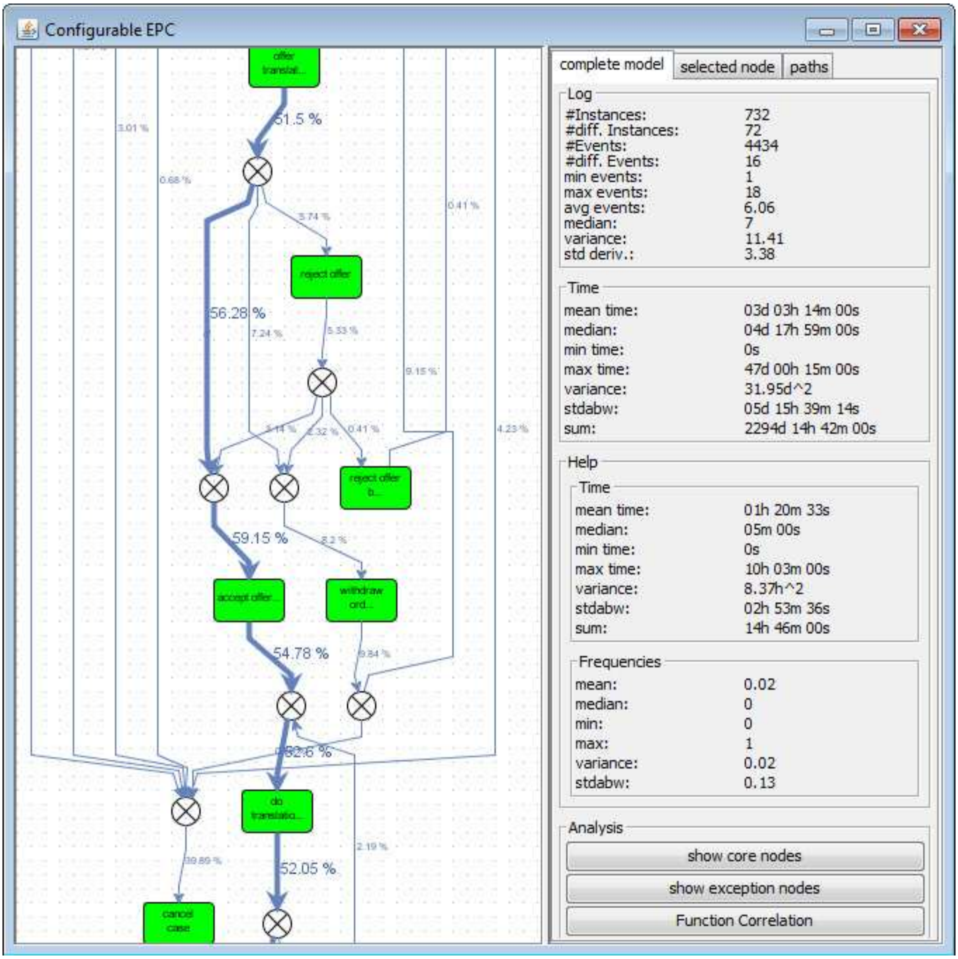

Figure 2: Process analysis with prototypical implementation in the RefMod-Miner

4.5 Recommendation Derivation

The recommendation derivation phase aims at interpreting the analysis results and delivering concrete hints concerning the process and 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 usability (e.g. avoided functionalities, needless undo-actions within process execution, misuse of functionalities, missing functionalities, unclear labels, very seldom or not used 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? E.g. some users first look for orders while others start with customer management. Are there significant differences in using a system?
- Are there observable case profiles for a process influencing its execution? E.g. minor vs. major order, standard delivery vs. transport of hazardous goods.
- Are there further functional requirements at the user side? E.g. working offline at permanent inventory since there is no network access available at some positions of a stockroom, direct access to the contact list instead of using external software.
- Are there possibilities to improve the process (e.g. user or case sensitive processing, adding new functionalities, data preloading, reorganization of forms)?

Input: Analysis results (e.g. metric values, correlations, etc.)

Content: Different approaches of automatic and manual data mining / interpretation techniques.

Output: Textual hints for improving the process or system usability, weak points, etc.

Running example: The above mentioned simple statistical information already provides an indication of weak points in a system. If e.g. the duration of an activity has a high variance, it could be the case that it cannot be handled intuitively. Users may have problems executing these tasks, thus, new employees will need much more time than troupers. Another reason might be a hidden function that is not supported by the BIS.

Considering the most probable paths, one observes that nearly 30% of the initiated translation cases, are being canceled immediately after displaying the automatically calculated translation costs. The next two paths cover successful process executions, whereby, within 17% of the cases, the automatically calculated translation costs are accepted. But in more than 21% of the cases, customers request an individual calculation. As it is not clear whether the reason for which the price could not be calculated automatically is the cost or the case, one should do both, analyze this system point and enrich the further logging strategy in order to log that aspect.

4.6 Implementation

The implementation phase covers the selection, design, planning and implementation of possible solutions for the afore generated hints. Against that background, this phase marks the beginning of a new lifecycle iteration.

Input: Textual hints (recommendations)

Content: Selection, design, planning and implementation of solutions for the identified improvement capabilities

Output: New system release

Running Example: This phase is not part of the running example, as the intention was to demonstrate the procedure identifying improvement capabilities.

5 Conclusion and Outlook

The paper at hand gives an overview of the concept of usability mining and constitutes different aspects which need to be investigated in order to be able to gather hints on the further development of a 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. In addition, both the existing metrics in the context of business process management and usability engineering and the named ideas should be analyzed concerning their information value for the mentioned context.

Hence, the usability mining approach creates the missing link between the software engineering view and the process-oriented view on BIS, which leads to promising potentials on the design and further development of business information systems. Hence, this dissertation is in the core of information systems research.

Furthermore, the intended 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. Thus, a deformation of measurement results is being obviated which is traditionally the case in 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. Particularly SME often collapse on the costs of existing alternatives [PR02].

References

[Aa08] 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.

[Aa11] van der Aalst, W.: Process Mining: Discovery, Conformance and Enchancement of Business Processes. Springer, Berlin, Heidelberg, 2011.

[AW05] van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering process models from event logs, 2005.

[BGT05] Balbo, S., Goschnick, S., Tong, D., Paris, C.: Leading Usability Evaluations to WAUTER. In: 11th Australian World Wide Web Conference, 2005.

[CW98] Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. ACM Transcations on Software Engineering and Methodology 7, 1998.

[EDG13] 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.

[HMP04] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 2004, pp. 75-105.

[Ho06] Hornbaeck, K.: Current practice in measuring usability: Challenges to usability studies and research. International Journal of Human-Computer Studies 64, 2006.
[HR99] Hilber, D.M., Redmiles, D.F.: Extracting Usability Information from User Interface Events. ACM Computing Surveys (CSUR) 32, 1999, pp. 384-421.
[IH01] 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.
[Ka06] Ana Karla, M.: Genetic Process Mining. CIP-Data Library Technische Universität Eindhoven, Eindhoven, 2006.
[Ka07] Kassem, G.: Application Usage Mining: Grundlagen und Verfahren. Shaker Verlag, Aachen, 2007.
[IS98] ISO: 9241: Ergonomic requirements for office work with visual display terminals. Part 11 : Guidance on usability, 1998.
[JA09] Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. Proceedings of the SIAM International Conference on Data Mining, SDM 2009, Sparks, Nevada, USA, 2009, pp. 401-412.
[JA10] 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.
[JJ13] Jareevongpiboon, W., Janecek, P.: Ontological approach to enhance results of business process mining and analysis. Business Process Management Journal 19, 2013.
[KLS09] 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.
[Me12] Melcher, J.: Process Measurement in Business Process Management – Theoretical Framework and Analysis of Several Aspects. KIT Scientific Publishing, Karlsruhe, 2012.
[MGG08]de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Saccà, D.: Process Mining Based on Clustering: A Quest for Precision. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) Business Process Management Workshops, LNCS 4928, Springer, Berlin, 2008, pp. 17-29.
[Ni93] Nielsen, J.: Usability Engineering. Academic Press, Boston, 1993.
[PR02] Peisner, M., Röse, K.: Usability Engineering in Germany: Situation, Current Practice and Networking Strategie. In: 1st European Usability Professionals Association Conf., 2002.
[SA08] Song, M., Günther, C.W., Van der Aalst, W.M.P.: Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops, LNBIP 17, Springer, Berlin, 2008, pp. 109-120.
[SDK06] 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.
[SE91] 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.
[Sh08] Shah, I.: Event Patterns as Indicators of Usability Problems. Journal of King Saud University – Computer and Information Sciences 20, 2008, pp. 31-43.
[Th13] Thaler, T.: Entwicklung einer Methode zum Process Mining unter besonderer Berücksichtigung von Organisationswissen. Semiramis Research and Service Unit, 2013.
[TK05] 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.
[TPL13] Thaler, T., Fettke, P., Loos, P.: Process Mining - Fallstudie leginda.de. HMD Praxis der Wirtschaftsinformatik 293, 2013, pp. 56-66.
[WAM06]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.
[WR11] Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible Heuristics Miner (FHM). In: IEEE Symposium on Computational Intelligence and Data Mining, 2011.