Exposé

Semantic Zooming

Name: Constantin Renner
Modul: Profilprojekt Anwendungsforschung in der Informatik (INF-PM-FPA)
Betreuer: Maximilian Weisenseel

1. Motivation

Today's business processes are a source of large amounts of data that are recorded during process execution and stored in event logs. Each event in the event log typically contains information about the timestamp at which an activity of the process was executed, the case in which it ourred, the resource involved and may include additional attributes such as measurements [1]. Depending on the underlying process, event logs can contain sensitive personal information. For example, in a healthcare context, an event log may contain information about patients (cases) who are treated by a physician (resource) during a specific activity and may include details about a disease as an additional event attribute [2]. Given sufficient background knowledge - for example a patient's activity trace or the timestamp at which a particular activity was executed - an attacker may be able to re-identify individuals and gather sensitive information about the patient. Privacy-preserving techniques are necessary to protect the cases from re-identification and from leaking sensitive information about an entity when the event log is published. At the same time, event logs are a fundamental starting point for process mining and other data-driven analyses for understanding and improving business processes [3]. This leads to a trade-off between preserving the privacy interests of individuals or entities represented in the event log and maintaining the utility of the data for analysis and insight generation [4].

2. Research Proposal and Related Work

To measure the privacy of personal data, various metrics such as k-anonymity, l-diversity, and t-closeness can be applied to a dataset [5]. Different techniques can be used to achieve anonymization and increase the privacy of datasets. These include the suppression of records, additive methods such as the insertion of noise, value swapping, or the generalization of attributes [6].

As a tool, ARX provides methods to ensure the privacy of a dataset regarding several privacy metrics like k-anonymity [2]. These approaches are primarily applied to relational data [7]. For event logs, additional information like the sequence of events (the trace) are present in the data and have to be protected as well. The trace itself can serve as an attack vector for identifying a case, while it may also contain sensitive information.

Several approaches for privacy-preserving publication of event logs have been proposed. PRIPEL is an approach that anonymizes an event log for publication based on differential privacy by inserting noise into a case's trace and its attributes [8]. PRETSA uses prefix trees to align traces to a common structure in order to achieve k-anonymity and t-closeness. Less frequent traces are transformed into more common ones, necessary event attributes are artificially generated [4]. TLKC-privacy is based on a k-anonymity approach that assumes an attacker can only determine a limited number of quasi-identifiers. (Sub-)traces are removed from the event log using a greedy approach if they violate privacy constraints.

These approaches have been implemented in various tools, such as PC4PM (inlcuding TLKC, PRIPEL) [1], PRIPEL in a tool described in [8], or anonymization approaches for classical relational data as implemented in ARX [2]. All of these approaches follow the principle of applying privacy-enhancing methods until a predefined privacy metric is satisfied. Their common goal is to preserve utility for process mining tasks like process discovery and to achieve results similar to those using the unmodified

event log. At the same time, they aim to prevent the re-identification of cases based on knowledge of events or traces.

The existing approaches focus on the identification of cases based on various properties, such as traces, case attributes, event attributes, or combinations of these. By applying different privacy-enhancing techniques, these attack vectors can be mitigated, preventing the re-identification of a case and, consequently, the extraction of its trace, case attributes, event attributes, and other sensitive information from the event log. In contrast, our approach aims to ensure that knowledge about a case, such as information about one or more events — i.e., the execution of a specific activity at a certain point in time — does not allow inference of the exact trace or other attributes of that case. Our approach proposes a drill-down that starts with a high level of privacy achieved through strong aggregation and gradually reduces it down to a defined threshold (until a privacy budget is exhausted). In this process, the user (and not an algorithm) decides which aspects should be less aggregated and, consequently, where privacy is reduced. The primary focus is on enabling the user to explore the aspects of the event log that are of interest like some specific activities or a specific event attribute, rather than on preserving optimal process mining results. To achieve this, event logs are initially published in a highly aggregated form. Users can then decide which parts of the event log they are interested in and progressively reduce the level of aggregation to obtain more detailed views of activities, cases, or other attributes. The cost to the user's privacy budget is calculated by applying privacy metrics to the resulting event log. As the level of detail increases and privacy protection decreases, the associated cost to the user correspondingly increases. A key technique in this approach is generalization. It allows users to first obtain an overview of the data and then reduce the level of generalization on demand for specific aspects. For the generalization or aggregation of attributes, these attributes must be organized into hierarchies. Such hierarchies can be syntactic (e.g., IP addresses) or semantic, which requires domain knowledge [9].

The objective of this project is therefore the development of a tool that allows users to examine self-selected, detailed parts of an event log while keeping the remaining parts in an aggregated state. The privacy of individual cases is preserved and the identification of cases and traces is prevented. The user can obtain a more detailed view of attributes (e.g., timestamps) or of selected activities, cases, time intervals, etc. Reducing the level of aggregation consumes the user's privacy budget. Aggregated events are to be visually presented to the user in a frontend, and interactive controls will allow the user to adjust the level of aggregation. The identification of individual cases should be prevented and verified using privacy metrics such as k-anonymity. A privacy budget and a cost function must be defined that meaningfully limit the amount of information a user can extract from the event log.

3. Approach

Starting from an event log, the user is initially presented a highly aggregated representation of the event log. The aggregation of events is performed based on their attributes, such that events are grouped into equivalence classes [9]. For this purpose, a generalization or aggregation rule must be defined for each attribute, ideally supporting multiple levels of aggregation. By categorizing attributes into different attribute classes, it may be possible to identify aggregation methods that can be applied to sets of attributes (e.g., syntactic hierarchies or for activities, the construction of a process model and its decomposition into sub-processes as a form of aggregation for activities [10]). The user can decide which attributes should be examined and visualized in more detail and can accordingly reduce the level of aggregation for those attributes. To limit the level and amount of detail that the user can extract from the event log, the user is assigned a privacy budget. Each zoom operation reduces this budget. By reducing the level of generalization of an attribute, k-anonymity decreases; the user's privacy budget is therefore consumed, while simultaneously providing a more detailed view of the corresponding aspect of the event log.

With an aggregation for an attribute applied, the case holds an aggregated value for this attribute. A user cannot determine which actual value it holds as several values are aggregated into the same aggregated value.

To be displayed to the user, a case must occur at least k times in the event log with respect to its trace,

in order to ensure k-anonymity at the trace level and reduce the risk of re-identification. To provide k-anonymity at the trace level, k traces must be indistinguishable. This requires considering both the attributes of the events within a trace and the transitions between events in that trace. Events are mapped to a level of abstraction based on the aggregation of time, activities, and other attributes, such that individual events become indistinguishable. For example, a concrete timestamp may be generalized to the day level. Two events occurring on the same day can then no longer be distinguished based on their timestamps. In addition to the events themselves, the transitions between events must also be taken into account. Overall, k traces must be indistinguishable: the sequence of events must be identical, and the individual events must not be distinguishable from one another. If fewer than k identical traces are present, the trace—and consequently the corresponding case—is not displayed. Based on the ensured k-anonymity, the cost to the user's privacy budget can be determined or a limit for reducing the aggregation can be enforced.

Beside the elimination of traces as described in [7], traces that occur infrequently can be merged with more frequent traces [4].

To obtain a detailed view of a specific attribute value, such as a particular activity, the user should be able to select that attribute value so that the aggregation level for this specific value is reduced. This allows a particular attribute for example a specific activity to be examined in greater detail without reducing the aggregation level of all activities. The remaining activities stay aggregated, resulting in higher overall privacy and lower consumption of the user's privacy budget.

For the visualization of the event log, the existing visualizations provided by SEAMLESS ZOOM[1] can be used as a foundation. The tool already supports the import of event logs in common formats such as CSV and XES, the visualization of event logs, and multiple abstraction levels. The event log visualization represents the execution of a case as a set of points in a time–activity diagram. These points are connected to visualize the trace of a case. The tool allows users to adjust the abstraction level between the instance level, where each case is shown with its activity execution over time, and the process level, where the process model is represented as a Directly-Follows Graph (DFG). Intermediate abstraction levels between these two extremes are also usable. These abstraction levels are generated using a kernel density estimator [11].

Building on this foundation, custom abstraction levels based on attribute aggregation must be implemented, along with zoom interactions for adjusting the abstraction level. Through abstraction, points are grouped in a manner similar to the existing tool, such that time intervals or aggregated activities are visualized as areas containing the original points and representing an equivalence class of events. For this purpose, a slider can be implemented for each attribute, allowing the user to interactively adjust the aggregation level of that attribute. Selecting a more detailed view will split the areas that contain the points representing the events into multiple equivalence classes visualized again as areas in the diagram. Other aspects besides time, case and activity like resource cannot be easily displayed in the 2D plot. Therefore a filter (e.g. via a dropdown) and the visualization of the filtered instances or an apporach based on color encoding must be established.

4. Skills

To determine the cost of a zoom operation, privacy metrics are employed. This requires familiarization with these metrics, with a primary focus on k-anonymity. In addition, further methods and techniques for data perturbation as well as the principles of differential privacy must be examined and acquired. For the implementation of the visualization, familiarization with the chosen framework and the corresponding programming language is required, depending on the selected approach. The usage of the existing tool[1] involves necessary familiarization with JavaScript and in particular the D3 library for extending the existing web application.

[1] `https://github.com/rubenssohn/SEAMLESS_ZOOM`

5. Schedule

The project is planned to span a period of six months, corresponding to approximately 31 weeks. During the initial weeks, an example event log will be used to investigate the generalization and aggregation of attributes. For this purpose, hierarchies for the different attributes must be developed, enabling the implementation of multiple aggregation levels. Starting in the third week, the computation and implementation of k-anonymity will be addressed, allowing a privacy metric to be associated with each aggregation level. Following a more in-depth familiarization with the visualization framework, the visualization of aggregated attributes is planned to be available by the end of week seven. Initially, the core attributes required for an event log (case, activity, timestamp) will be considered. In parallel further datasets will be employed to test the newly developed components. During the subsequent two weeks, the privacy budget available to the user for exploring the dataset will be defined. Furthermore, a function for calculating the cost of zoom operations based on privacy metrics (in particular k-anonymity) must be developed. From week ten onward, attributes beyond the classical event log attributes will be examined. These attributes must also be visualized, aggregated and zooming within their aggregation levels must be supported. In addition to reducing the aggregation level of an attribute, the system should enable the inspection of specific activities, cases, time intervals and similar aspects. For these aspects, cost functions with respect to the privacy budget, based on k-anonymity, must likewise be defined. The implementation of the corresponding visualizations is planned to take five weeks. Further aspects to be implemented at least prototypically include data perturbation in cases where k-anonymity can no longer be guaranteed by the available cases. This requires the investigation and implementation of mechanisms for generating perturbed data, as well as the definition of their associated privacy budget costs, for which four weeks are allocated. An additional week is reserved for any necessary adaptations of the visualization based on these extensions. Towards the end of the project, the documentation will be finalized and the project presentation will be prepared.

References

[1] M. Rafiei, A. Schnitzler, and W. M. P. v. d. Aalst, *PC4PM: A Tool for Privacy/Confidentiality Preservation in Process Mining*, en, arXiv:2107.14499 [cs], Jul. 2021. DOI: 10.48550/arXiv.2107.14499. [Online]. Available: http://arxiv.org/abs/2107.14499 (visited on 12/20/2025).

[2] O. Kamal, S. Sohail, and F. Bukhsh, "Optimizing Privacy-Utility Trade-Off in Healthcare Processes: Simulation, Anonymization, and Evaluation (Using Process Mining) of Event Logs:" en, in *Proceedings of the 14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Dijon, France: SCITEPRESS - Science and Technology Publications, 2024, pp. 289–296, ISBN: 978-989-758-708-5. DOI: 10.5220/0012766800003758. [Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0012766800003758 (visited on 12/20/2025).

[3] F. Mannhardt, A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael, "Privacy-Preserving Process Mining: Differential Privacy for Event Logs," en, *Business & Information Systems Engineering*, vol. 61, no. 5, pp. 595–614, Oct. 2019, ISSN: 2363-7005, 1867-0202. DOI: 10.1007/s12599-019-00613-3. [Online]. Available: http://link.springer.com/10.1007/s12599-019-00613-3 (visited on 12/18/2025).

[4] S. A. Fahrenkrog-Petersen, H. Van Der Aa, and M. Weidlich, "PRETSA: Event Log Sanitization for Privacy-aware Process Discovery," en, in *2019 International Conference on Process Mining (ICPM)*, Aachen, Germany: IEEE, Jun. 2019, pp. 1–8, ISBN: 978-1-7281-0919-0. DOI: 10.1109/ICPM.2019.00012. [Online]. Available: https://ieeexplore.ieee.org/document/8786060/ (visited on 12/25/2025).

[5] D. Hauf, "K-Anonymity, l-Diversity and T-Closeness," de,

[6] M. Rafiei and W. M. P. Van Der Aalst, "Privacy-Preserving Data Publishing in Process Mining," en, in *Business Process Management Forum*, D. Fahland, C. Ghidini, J. Becker, and M. Dumas, Eds., vol. 392, Series Title: Lecture Notes in Business Information Processing, Cham: Springer International Publishing, 2020, pp. 122–138, ISBN: 978-3-030-58637-9 978-3-030-58638-6. DOI: 10.1007/978-3-030-58638-6_8. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58638-6_8 (visited on 12/22/2025).

[7] M. Rafiei, M. Wagner, and W. M. P. Van Der Aalst, "TLKC-Privacy Model for Process Mining," en, in *Research Challenges in Information Science*, F. Dalpiaz, J. Zdravkovic, and P. Loucopoulos, Eds., vol. 385, Series Title: Lecture Notes in Business Information Processing, Cham: Springer International Publishing, 2020, pp. 398–416, ISBN: 978-3-030-50315-4 978-3-030-50316-1. DOI: 10.1007/978-3-030-50316-1_24. [Online]. Available: http://link.springer.com/10.1007/978-3-030-50316-1_24 (visited on 12/20/2025).

[8] S. A. Fahrenkrog-Petersen, H. Van Der Aa, and M. Weidlich, "PRIPEL: Privacy-Preserving Event Log Publishing Including Contextual Information," en, in *Business Process Management*, D. Fahland, C. Ghidini, J. Becker, and M. Dumas, Eds., vol. 12168, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 111–128, ISBN: 978-3-030-58665-2 978-3-030-58666-9. DOI: 10.1007/978-3-030-58666-9_7. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58666-9_7 (visited on 12/20/2025).

[9] R. Hildebrant, S. A. Fahrenkrog-Petersen, M. Weidlich, and S. Ren, "PMDG: Privacy for Multi-perspective Process Mining Through Data Generalization," en, in *Advanced Information Systems Engineering*, M. Indulska, I. Reinhartz-Berger, C. Cetina, and O. Pastor, Eds., vol. 13901, Series Title: Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2023, pp. 506–521, ISBN: 978-3-031-34559-3 978-3-031-34560-9. DOI: 10.1007/978-3-031-34560-9_30. [Online]. Available: https://link.springer.com/10.1007/978-3-031-34560-9_30 (visited on 12/20/2025).

[10] S. C. De Alvarenga, S. Barbon, R. S. Miani, M. Cukier, and B. B. Zarpelão, "Process mining and hierarchical clustering to help intrusion alert visualization," en, *Computers & Security*, vol. 73, pp. 474–491, Mar. 2018, ISSN: 01674048. DOI: 10.1016/j.cose.2017.11.021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404817302584 (visited on 12/13/2025).

[11] C. Rubensson and J. Mendling, "Time-Order Map for Seamless Zooming between Process Models and Process Instances," en, in *2025 7th International Conference on Process Mining (ICPM)*, Montevideo, Uruguay: IEEE, Oct. 2025, pp. 1–8, ISBN: 979-8-3315-8596-9. DOI: 10.1109/ICPM66919.2025.11220736. [Online]. Available: https://ieeexplore.ieee.org/document/11220736/ (visited on 12/11/2025).